DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943-5002

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AN AUTOMATIC CONTROL DESIGN
FOR
THE MARINER CLASS SHIPS

by

Tayfun Tansan

December 1984

Thesis Advisor: George J. Thaler

Approved for public release; distribution is unlimited

T224022

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

An Automatic Control Design
for
the Mariner Class Ships

5. TYPE OF REPORT & PERIOD COVERED
Master's thesis
December 1984

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Tayfun Tansan

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate Scholl
Monterey, California 93943

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93943

12. REPORT DATE
December 1984

13. NUMBER OF PAGES
99

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Function Minimization, The Cost Function, The Mariner Class Ship's Equation of Motions, NOMOTO Equations.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The increase in fuel prices has initiated considerable interest by ship operators in new ship autopilots which minimize the propulsion losses due to steering.

This research presents the results of work on a steering design for the Mariner class ship based on a computer simulation. A model of the Mariner class ship was coupled to a function minimization subroutine to minimize the added resistance (cont.)

ABSTRACT (Continued) # 20

caused by rudder activity and hull drag of inertial origins caused by periodic yawing of ship in seaway.

The Mariner class ship computer model was tested in calm water and in seaway. The optimal controller parameters are shown in look up tables as functions of ship speed, sea state, encounter angle and encounter frequency. This technique can be used as an adaptive controller.

Approved for public release; distribution is unlimited.

An Automatic Control Design for the Mariner Class Ships

by

Tayfun Tansan
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1984

ABSTRACT

The increase in fuel prices has initiated considerable interest by ship operators in new ship autopilots which minimize the propulsion losses due to steering.

This research presents the results of work on a steering design for the Mariner class ship based on a computer simulation. A model of the Mariner class ship was coupled to a function minimization subroutine to minimize the added resistance caused by rudder activity and hull drag of inertial origins caused by periodic yawing of ship in seaway.

The Mariner class ship computer model was tested in calm water and in a seaway. The optimal controller parameters are shown in look up tables as functions of ship speed, sea state, encounter angle and encounter frequency. This technique can be used as an adaptive controller.

TABLE OF CONTENTS

I. INTRODUCTION 11

II. NONLINEAR EQUATIONS OF MOTION 13

III. OPEN LOOP SHIP'S BEHAVIOR IN CALM WATER AND SOME SEA STATES 20

A. CALM WATER CASE 20

B. SEA STATE CASE 20

IV. NOMOTO MODEL OF THE MARINER 24

A. MATHEMATICAL APPROACH 24

B. COMPUTER APPROACH 25

V. DERIVATION OF A COST FUNCTION FOR THE MARINER SHIP 28

VI. CONTROLLER DESIGN FOR THE MARINER, USING FORTRAN PROGRAM 34

A. CALM WATER CASE 34

B. REGULAR SEAS CASE 36

VII. AN APPROACH TO AN ADAPTIVE AUTOPILOT 64

VIII. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE STUDY 67

A. CONCLUSIONS 67

B. RECOMMENDATIONS FOR FUTURE STUDY 68

APPENDIX A: THESIS FORTRAN 70

APPENDIX B: DATA FOR SEA STATE PROGRAM 74

APPENDIX C: PROGRAM TO CALCULATE OPTIMAL GAINS 78

LIST OF REFERENCES 96

INITIAL DISTRIBUTION LIST 99

LIST OF TABLES

I. Assessment of the Coefficients in the X_Equation 17
II. Assessment of the Coefficients in the Y_Equation 18
III. Assessment of the Coefficients in the N_Equation 19
IV. The Nomoto Model Parameters for Mariner 27
V. Values of Weighting Factor for Different Speeds of The Mariner Class Ship 33
VI. Optimal Controller Parameters I 35
VII. Optimal Controller Parameters II 35
VIII. Optimal Controller Parameters III 35
IX. Optimal Controller Parameters IV 36
X. Optimal Controller Parameters V 37
XI. Optimal controller Parameters VI 37
XII. Optimal Controller Parameters VII 37
XIII. Optimal Controller Parameters VIII 38
XIV. Optimal Controller Parameters IX 38
XV. Optimal Controller Parameters X 38
XVI. Optimal Controller Parameters XI 39
XVII. Optimal Controller Parameters XII 39
XVIII. Location of the Ship 47
XIX. Test for LAMBDA Value I 49
XX. Test for LAMBDA Value II 50
XXI. Sea State vs Range for Wave Height 78
XXII. Sea State vs Wave Height 79

LIST OF FIGURES

3.1 Time Response of U, V AND R in Calm Water when D = 1 Degree 22
3.2 Time Response of U,V and R in Regular Seas 23
4.1 Determination of Third and Second Order Nomoto Models 26
6.1 Various Structure Controllers 34
6.2 Rudder Angles in Degrees for Sea State 6 and 8 . . 40
6.3 Heading Errors in Degrees for Sea State 6 and 8 . . 41
6.4 The Cost vs. K 42
6.5 The Cost vs. Z1 43
6.6 The Cost vs. P1 44
6.7 The Cost vs. Z1 and P1 45
6.8 Orientation of Space Axes and Moving Axes 46
6.9 Simulation Results-Steady State 600 sec 48
6.10 The Cost Curves vs. Z1 and P1 when Parameters are Changing Around the Optimal Values for the Calm Water Case 51
6.11 The Cost Curves vs. Z1 and P1 when Parameters Changing Around the Optimal Values for the Sea State Case 52
6.12 Open Loop Steering Model 54
6.13 Open Loop System BODE and NICHOLS Plots. (Using Comp.'A') 55
6.14 Open Loop System BODE and NICHOLS Plots. (Using Comp.'B') 56
6.15 Rudder Motion in Case I with Case I and Case II Parameters 58
6.16 Heading Error in Case I with Case I and Case II Parameters 59

6.17 Rudder Motion in Case II with Case II and Case I Parameters 59
6.18 Heading Error in Case II with Case II and Case I Parameters 60
6.19 Rudder Motion in Case III with Case III and Case I Parameters 60
6.20 Heading Error in Case III with Case III and Case I Parameters 61
6.21 Rudder Motion in Case I with Case I and Case III Parameters 61
6.22 Heading Error in Case I with Case I and Case III Parameters 62
7.1 Adaptive Control Scheme 65
C.1 Algorithm to Calculate Optimal Gains 77

ACKNOWLEDGMENT

First I would like the thank the Turkish Navy and the United States Navy and the special relationship between these navies that permitted the completion of my research. I want to thank Lt. Vincente Garcia for his cooperation and teamwork that existed throughout my thesis project was most essential to its final completion. And, most important, I want to thank Professor George J. Thaler for his invaluable assistance.

I. INTRODUCTION

Claims by many researchers indicate that a carefully designed controller could reduce fuel consumption by minimizing the propulsion losses which are caused by added drag due to steering of the ship.

The goal of this thesis was aimed at developing and demonstrating the utility of an improved steering control for the Mariner class ship. The immediate goal of this research was to develop design methodology for an adaptive autopilot that would provide effective steering control with associated cost savings for a full range of seaway and stability conditions.

To simulate the ship in the computer program, ship's nonlinear equations of motion were needed. Chapter 2 addresses the Mariner class ship nonlinear equations of motion.

Chapter 3 addresses the work on testing the ship simulation model and open loop ship's behaviors in calm water and in a seaway.

The basic Nomoto models give an adequate description of ship steering dynamics for design. The Nomoto second and third order models were developed from the ship's linear equations of motion, Chapter 4 adresses the Mariner class ship Nomoto model derivations by using mathematical methods and by using a function minimization subroutine.

Chapter 5 shows an adequate cost function which represents the added drag due to steering and includes derivations for evaluating the weighting factor. Also Chapter 5

presents the assumptions and approaches needed to find the cost function that is used by many researchers.

Ship dynamics change with operating conditions such as ship speed, encounter angle and encounter frequency. Chapter 6 presents optimal controller parameters as a function of different operating conditions.

Chapter 7 addresses an approach to an adaptive controller utilizing information which is easy to measure on ship board such as ship speed, heading error and rudder angle. This adaptive controller must be used to provide minimum added drag due to steering.

Conclusions were drawn from simulation results and are presented in Chapter 8. This chapter also recommends some future studies, which can be done as extensions of this work.

II. NONLINEAR EQUATIONS OF MOTION

Nonlinear equations of motion are suitable for predicting tight maneuvers and also suitable for computer programming. The nonlinear equations of motion used in this work have been developed by Abkowitz [Ref. 1, 2], and Strom_Tejsen [Ref. 3], based on a Taylor series expansion of forces and moments. Terms higher than third order are not included in the equations because experience has shown that accuracy is not significantly improved by their inclusion.

A result of symmetry about the xz-plane, X is an even function of V, R, D, $\dot{V}$ and $\dot{R}$ so on, the crosscoupled terms in the equations involving odd powers of V, R and D are zero, however, crosscouple terms which involve even powers of V,R and D are nonzero. In contrast to X, the expressions for Y and N are odd functions of V,R,D,$\dot{V}$ and $\dot{R}$; that is, only the coefficients of the terms in the expansion with odd powers are nonzero; those with even powers are zero. For some reasons, X is neither an odd nor an even function of U but rather its expansion includes all powers of DU.

Equations X,Y and N are functions of U,V,R,$\dot{U}$,$\dot{V}$,$\dot{R}$ and D. Taylor series expansions of X,Y and N including terms up to the third order are as follows:

$$(m - X_{\dot{U}})*\dot{U} = f_1(U,V,R,D) \qquad \text{(eqn 2.1)}$$

$$(m - Y_{\dot{V}})*\dot{V} + (m*X_G - Y_{\dot{R}})*\dot{R} = f_2(U,V,R,D) \qquad \text{(eqn 2.2)}$$

$$(m*X_G - N_{\dot{V}})*\dot{V} + (I_Z - N_{\dot{R}})*\dot{R} = f_3(U,V,R,D) \qquad \text{(eqn 2.3)}$$

Where:

$X_{uu} = d^2X/dU^2$ $\quad X_{DD} = d^2X/dD^2$ $\quad X_{uv} = d^2X/dU*dV$

$X_{uuu} = d^3X/dU^3$ $\quad X_{uuR} = d^3X/dU^2*dR$ $\quad$ etc.

$$f_1(U,V,R,D) = 0.5*X_{uu}*DU^2 + (1/6)*X_{uuu}*DU^3 + 0.5*X_{vv}*V^2 + (0.5*X_{RR} + m*X_G)*R^2 + 0.5*X_{vvu}*V^2*DU + 0.5*X_{Rru}*R^2*DU + (X_{vR} + m)*V*R + X_{vD}*V*D + X_{vru}*V*R*DU + X_{vDu}*V*D*DU + X_{RDu}*R*D*DU + X_u*DU + X^0 + 0.5*X_{DD}*D^2 + 0.5*X_{DDu}*D^2*DU + X_{RD}*R*D \quad \text{(eqn 2.4)}$$

$$f_2(U,V,R,D) = Y^0_u*DU + Y^0_{uu}*DU^2 + Y_v*V + Y_D*D + Y^0 + 0.5*Y_{vRR}*V*R^2 + 0.5*Y_{vDD}*V*D^2 + Y_{vu}*V*DU + 0.5*Y_{vuu}*V*DU^2 + (Y_R - m*U1)*R + (1/6)*Y_{DDD}*D^3 + Y_{Ru}*R*DU + 0.5*Y_{Ruu}*R*DU^2 + (1/6)*Y_{RRR}*R^3 + 0.5*Y_{Dvv}*D*V^2 + 0.5*Y_{Rvv}*R*V^2 + 0.5*Y_{DRR}*D*R^2 + Y_{Du}*D*DU + 0.5*Y_{Duu}*D*DU^2 + Y_{vRD}*V*R*D + (1/6)*Y_{vvv}*V^3 + 0.5*Y_{RDD}*R*D^2 \quad \text{(eqn 2.5)}$$

$$f_3(U,V,R,D) = N^0_u*DU + N^0_{uu}*DU^2 + N_v*V + N_D*D + 0.5*N_{Duu}*D*DU^2 + (N_R - m*X_G U1)*R + N_{vu}*V*DU + 0.5*N_{vuu}*V*DU^2 + N_{vRD}*V*R*D + N^0 + 0.5*N_{Dvv}*D*V^2 + 0.5*N_{DRR}*D*R^2 + N_{Du}*D*DU + N_{Ru}*R*DU + 0.5*N_{Ruu}*R*DU^2 + (1/6)*N_{DDD}*D^3 + (1/6)*N_{RRR}*R^3 + 0.5*N_{Rvv}*R*V^2 + 0.5*N_{RDD}*R*D^2 + (1/6)*N_{vvv}*V^3 + 0.5*N_{vRR}*V*R^2 + 0.5*N_{vDD}*V*D^2 \quad \text{(eqn 2.6)}$$

All of the derivative coefficients of the equations are evaluated on the basis of experimental data obtained from captive model tests, and given in Table I, II and III.

The Y force and N moment induced by the rotation of a single propeller or by unirotating multiple propellers at

V=D=0.0 are identified as Y° and N°, these are likely to be speed dependent. To see the rudder effects in calm water or sea state propeller effects were ignored.

Finally, from the X , Y and N equations the ship's surge, sway and yaw equations can be written as follows.

$$\dot{U} = \frac{f_1(U,V,R,D)}{(m - X_{\dot{U}})} \qquad \text{(eqn 2.7)}$$

$$\dot{V} = \frac{(I_z - N_{\dot{R}})*f_2(U,V,R,D) - (m*X_G - Y_{\dot{R}})*f_3(U,V,R,D)}{(m - Y_{\dot{V}})*(I_z - N_{\dot{R}}) - (m*X_G - Y_{\dot{R}})} \qquad \text{(eqn 2.8)}$$

$$\dot{R} = \frac{(m - Y_{\dot{V}})*f_3(U,V,R,D) - (m*X_G - N_{\dot{V}})*f_2(U,V,R,D)}{(m - Y_{\dot{V}})*(I_z - N_{\dot{R}}) - (m*X_G - Y_{\dot{R}})} \qquad \text{(eqn 2.9)}$$

These surge, sway and yaw equations can be rewritten in the form:

$$dU/dt = g_1(t, U(t), V(t), R(t), D(t))$$

$$dV/dt = g_2(t, U(t), V(t), R(t), D(t)) \qquad \text{(eqn 2.10)}$$

$$dR/dt = g_3(t, U(t), V(t), R(t), D(t))$$

Where U(t), V(t), R(t) and D(t) are the instantaneous values of U, V, R and D at any time t.

Equation 2.10 is a set of three first-order differential equations for which approximate numerical solutions are readily obtained on a digital computer. The key to the numerical solution is that values of U, V and R at time t+DELT are obtained from knowledge of the values of U, V, R and D at time t using a simple first-order expansion; that is,

$$U(t+DELT) = U(t) + DELT*U(t)$$

$$V(t+DELT) = V(t) + DELT*V(t) \qquad \text{(eqn 2.11)}$$

$$R(t+DELT) = R(t) + DELT*R(t)$$

This method is found to give adequate accuracy for the present type of differential equations because of the fact that the accelerations $\dot{U}$, $\dot{V}$ and $\dot{R}$ vary but slowly with time, due to the large mass and inertia of a ship compared to the relatively small forces and moments produced by its control surface. Any desired accuracy of the solutions can be obtained with a computer by simply using smaller time intervals DELT. This procedure was used for all computer programs which were developed for this thesis.

TABLE I

Assessment of the Coefficients[1] in the X_Equation

Variable	Coefficient	Value of Coefficient
$\dot{U}$	$m - X_{\dot{u}}$	0.177
DU	X_u	-0.0253
DU^2	$0.5*X_{uu}$	0.00948
DU^3	$(1/6)*X_{uuu}$	-0.00217
V^2	$0.5*X_{vv}$	-0.189
R^2	$0.5*X_{RR} + m*X_G$	0.00379
D^2	$0.5*X_{DD}$	-0.02
V^2*DU	$0.5*X_{vvu}$	–
R^2*DU	$0.5*X_{RRu}$	–
D^2*DU	$0.5*X_{DDu}$	–
V*R	$X_{VR} + m$	0.168
V*D	X_{VD}	0.0196
R*D	X_{RD}	0
V*R*DU	X_{VRu}	–
V*D*DU	X_{VDu}	–
R*D*DU	X_{RDu}	–
–	X^0	0

[1]All derivatives are nondimensionalized on the basis of RHO,L,T and S.
No entry in these columns means the coefficient was ignored.

TABLE II

Assessment of the Coefficients in the Y_Equation

Variable	Coefficient	Value of Coefficient
$\dot{V}$	$(m - Y_{\dot{v}})$	0.327
$\dot{R}$	$(m*X_G - Y_{\dot{R}})$	-0.0018
V	Y_v	-0.244
V^3	$(1/6)*Y_{vvv}$	-1.702
$V*R^2$	$0.5*Y_{vRR}$	0
$V*D^2$	$0.5*Y_{vDD}$	-0.0008
$V*DU$	Y_{vu}	–
$V*DU^2$	$0.5*Y_{vuu}$	–
R	$(Y_R - m)$	-0.105
R^3	$(1/6)*Y_{RRR}$	0
$R*V^2$	$0.5*Y_{Rvv}$	3.23
$R*D^2$	$0.5*Y_{RDD}$	0
$R*DU$	Y_{Ru}	–
$R*DU^2$	$0.5*Y_{Ruu}$	–
D	Y_D	0.0586
D^3	$(1/6)*Y_{DDD}$	-0.00975
$D*V^2$	$0.5*Y_{Dvv}$	0.25
$D*R^2$	$0.5*Y_{DRR}$	0
$D*DU$	Y_{Du}	0
$D*DU^2$	$0.5*Y_{Duu}$	–
$V*R*D$	Y_{vRD}	0
–	Y^0	-0.0008
DU	Y^0_u	0
DU^2	Y^0_{uu}	–

TABLE III

Assessment of the Coefficients in the N_Equation

Variable	Coefficient	Value of Coefficient
$\dot{V}$	$(m*X_G - N_{\dot{V}})$	-0.00478
$\dot{R}$	$(I_Z - N_{\dot{R}})$	0.0175
V	N_V	-0.0555
V^3	$(1/6)*N_{VVV}$	0.345
$V*R^2$	$0.5*N_{VRR}$	0
$V*D^2$	$0.5*N_{VDD}$	0.00264
$V*DU$	N_{VU}	–
$V*DU^2$	$0.5*N_{VUU}$	–
R	$(N_R - m*X_G*U1)$	-0.0349
R^3	$(1/6)*N_{RRR}$	0
$R*V^2$	$0.5*N_{RVV}$	-1.158
$R*D^2$	$0.5*N_{RDD}$	0
$R*DU$	N_{RU}	–
$R*DU^2$	$0.5*N_{RUU}$	–
D	N_D	-0.0293
D^3	$(1/6)*N_{DDD}$	0.00482
$D*V^2$	$0.5*N_{DVV}$	0.1032
$D*R^2$	$0.5*N_{DRR}$	0
$D*DU$	N_{DU}	0
$D*DU^2$	$0.5*N_{DUU}$	–
$V*R*D$	N_{VRD}	0
–	N^0	0.00059
DU	N^0_U	0
DU^2	N^0_{UU}	–

III. OPEN LOOP SHIP'S BEHAVIOR IN CALM WATER AND SOME SEA STATES

A. CALM WATER CASE

Using a ship's nonlinear equations of motion and Mariner class ship coefficients, a simulation program THESIS FORTRAN was developed and run to observe U, V and R.The computer program THESIS FORTRAN is shown in Appendix A.

First run D=0, YAWC=0, U1=15 knots were applied and it was seen that the ship stays on its initial course and speed. U=15 knots, V=R=0.

The program was rerun a few times, changing the rudder angle to 2.8 degrees to both sides (port and starboard) and it was observed that increasing the rudder angle changes the ship's course and also U decreased, the absolute values of V and R increased. After a few hundred seconds U, V and R reached steady values independently. These steady values depend on rudder angle, large rudder angles decrease U and increase V and R. For a rudder angle of one degree and constant speed (U1) of 15 knots, the first 200 seconds of time response of U, V and R are shown in figure 3.1 as an example.

B. SEA STATE CASE

To observe the behavior of the ship in a sea state, disturbance forces and moments are needed that depend on sea state, ship speed, encounter angle and encounter frequency. Also in sea steate hydrodynamic parameters are changed, i.e, the added mass and added inertia are functions of encounter frequency and sea state.

The disturbance forces, moments, and added mass, added inertia terms were found by running the sea state program that is presented in [Ref. 4]. Data for a Mariner class ship and chosen sea conditions that were used in the sea state program are shown in APPENDIX B.

For observation purposes 'FX' (disturbance force in surge), 'FY' (disturbance force in sway) and 'MZ' (disturbance moment in yaw) were added into the surge, sway and yaw equations that were used in the simulation program.

For regular seas the program has been run a few times.Every time different FX, FY, MZ and coefficients were used to represent different ship characteristics such as ship speed, loading [2] etc. and enviromentel conditions such as sea state, encounter angle, encounter frequency. Results show that U, V and R are sine waves with amplitude and phase depending on ship and environmental conditions.

Time response of U, V and R in 200 seconds are presented in Figure (3.2), for ship speed 15 knots, encounter angle 030.0 degree, encounter frequency 0.60 radian/second and sea state 6.

---

[2]During this research, displacements (Molded) up to 32 ft were chosen as a loading condition for the Mariner.

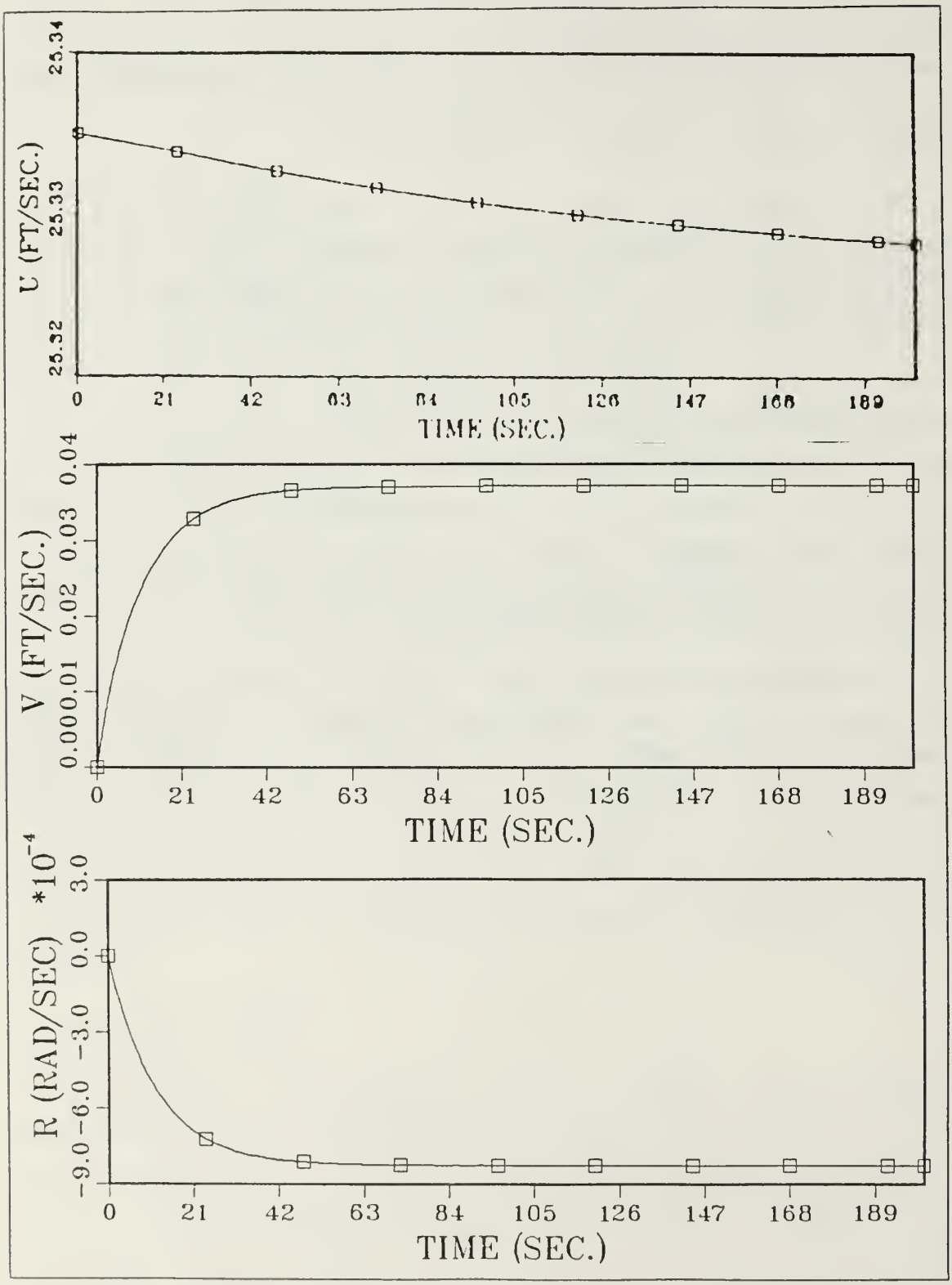


Figure 3.1 Time Response of U, V AND R in Calm Water when D = 1 Degree.

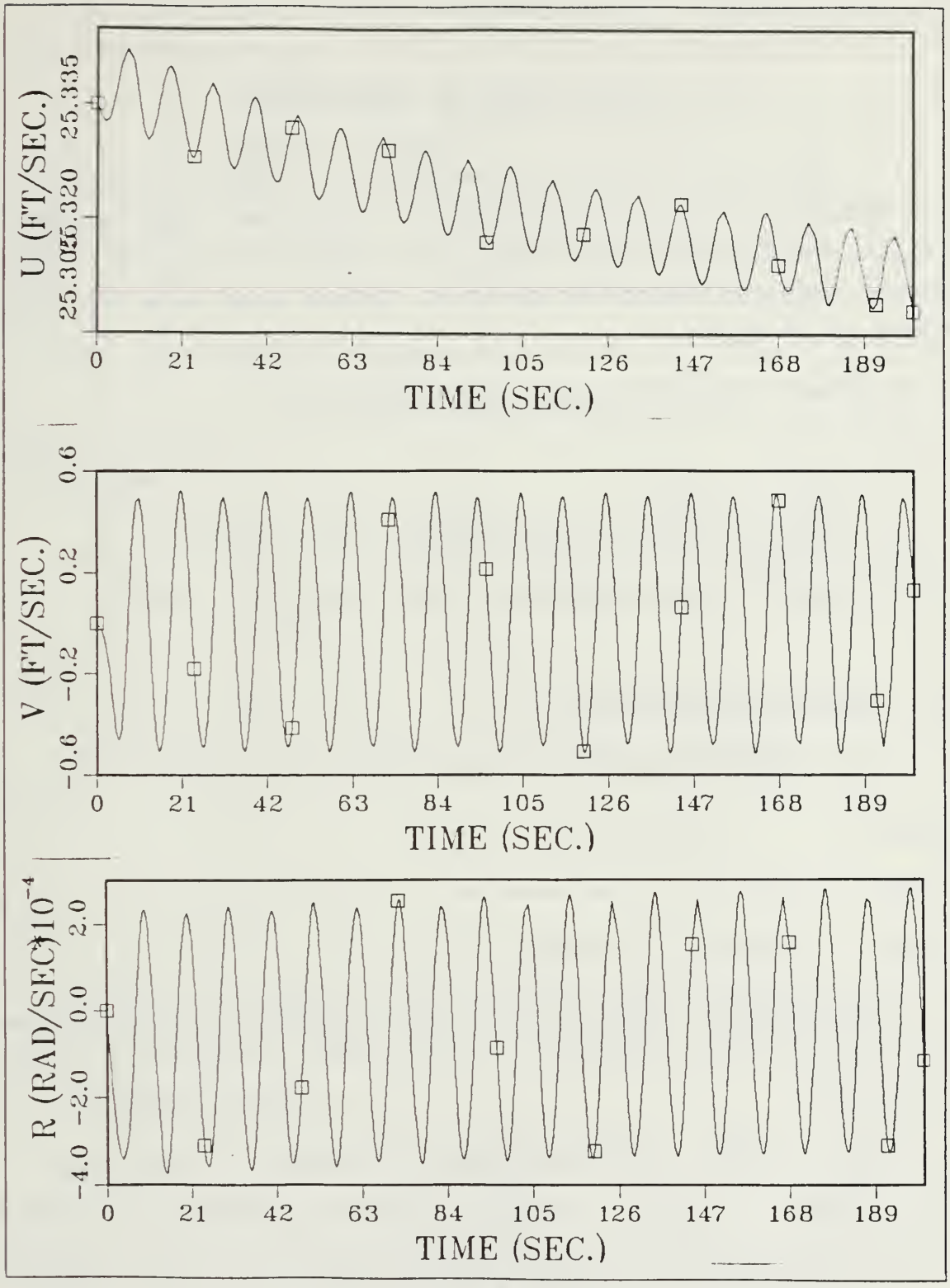


Figure 3.2 Time Response of U,V and R in Regular Seas.

IV. NOMOTO MODEL OF THE MARINER

To find a model which can be used in computer simulation the Mariner's linear equations of motion and its hydrodynamic coefficients were used and third and second order Nomoto transfer functions derived. Values used were for ship speed of 15 knots.

Mariner's linear equations of motion are

$$(m - X_{\dot{u}})*\dot{U} = X_u*DU \qquad \text{(eqn 4.1)}$$

$$(m - Y_{\dot{v}})*\dot{V} - Y_V*V = (Y_{\dot{R}} - m*X_G)*\dot{R} + (Y_R - m*U1)*R$$

$$(I - N_{\dot{R}})*\dot{R} - (N_R - m*X_G*U1)*R = (N_{\dot{v}} - m*X_G)*\dot{V} + N_V*V$$

A. MATHEMATICAL APPROACH

Proceeding to the third order Nomoto equation:

$$\frac{YAW(s)}{D(s)} = \frac{K*(Z*s + 1)}{s*(P1*s + 1)*(P2*s + 1)}$$

Deriving the third order Nomoto transfer function from the sway and yaw equations, we show them in matrix notation as follows.

$$\begin{bmatrix} V \\ R \end{bmatrix} = \begin{bmatrix} -.0372 & -8.42 \\ -.0003 & -.10 \end{bmatrix} * \begin{bmatrix} V \\ R \end{bmatrix} + \begin{bmatrix} .210 \\ -.003 \end{bmatrix} *D$$

X = A*X + B*U then, X(s) = $(s*I-A)^{-1}*B*U$ (assuming initial conditions are zero.)

$$\begin{bmatrix} V \\ R \end{bmatrix} = \begin{bmatrix} .2101*s + .0534 \\ -.0038*s - .0002 \end{bmatrix} * \frac{D}{(7.67*s+1)*(116.93*s+1)}$$

R(s) = s*YAW(s)

so, then $$\frac{YAW(s)}{D(s)} = \frac{-0.189*(18.34*s+1)}{s*(7.67*s+1)*(116.93*s+1)}$$

result is K=0.189, z=18.34, P1=7.67 and P2=116.93

Proceeding to the second order Nomoto equation:

$$\frac{YAW(s)}{D(s)} = \frac{K}{s*(P1*s + 1)}$$

Deriving the second order Nomoto transfer function from the yaw equation only, the result is K=0.03 and P1=10.

B. COMPUTER APPROACH

We used a function minimization subroutine to obtain parameters of the transfer functions. Figure 4.1 shows the scheme used to obtain the third and second order Nomoto transfer functions. The results are given in Table IV.

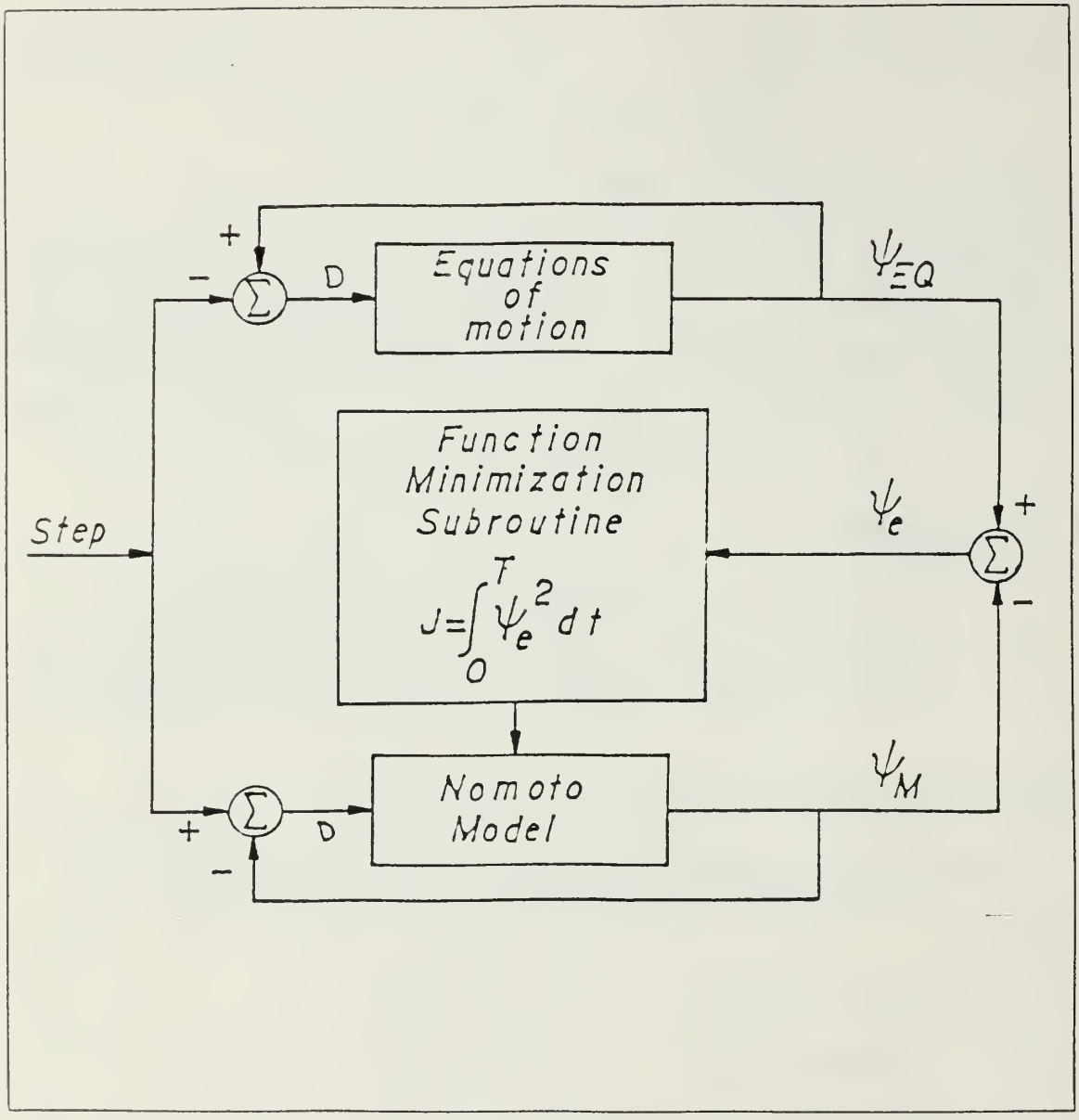


Figure 4.1 Determination of Third and Second Order Nomoto Models.

TABLE IV

The Nomoto Model Parameters for Mariner

Third Order	Second Order
K = 0.189	K =0.0298
Z = 18.347	P1 = 9.989
P1 = 7.6739	
P2 = 116.929	

As seen the answers obtained by function minimization agree closely with the analytic solutions.

V. DERIVATION OF A COST FUNCTION FOR THE MARINER SHIP

In recent years many researchers have studied the problem of optimizing an automatic ship steering controller for minimum fuel consumption. It is well known that additional drag is introduced by steering and that both the rudder motion and the yawing motion contribute to this added drag.

When deriving a cost function we are required to find one cost function that must be convenient for ship board use.The cost function that is commonly used in recent years is

$$J = \int_0^{t_o} (LAMDA*YAWE^2 + D^2)*dt \qquad \text{(eqn 5.1)}$$

Where YAWE = Yaw error
LAMBDA = Weighting factor

Derivation of this cost function from the surge equation has been well explained in [Ref. 5] for the SL/7 class ship by R.E.REID.

To derive the cost function for the Mariner, REID's approach is taken as a reference and his assumptions are used. To show how the cost function for the Mariner was derived, REID's work is presented here step by step with derivations of the Mariner's cost function.

The Surge Equations:

SL/7:

$$(m - X_{\dot{u}})*\dot{U} = X^0 + 0.5*X_{vv}*V^2 + 0.5*X_{DD}*D^2 + (X_{vR} + m)*V*R + Xp \qquad \text{(eqn 5.2)}$$

Where: X^0 = -0.0003 for SL/7
Xp = Propeller thrust

MARINER:

$$(m - X_{\dot{u}})*\dot{U} = 0.5*X_{vv}*V^2 + (X_{vR} + m)*V*R + 0.5*X_{DD}*D^2 + (0.5*X_{RR} + m*X_G)*R^2 + X_{vD}*V*D + X_u*DU + 0.5*X_{uu}*DU^2 + (1/6)*X_{uuu}*DU^3 \quad \text{(eqn 5.3)}$$

It is seen that there are some terms in the Mariner's surge equation which they are not included in the SL/7's surge equation. Assuming steady state situations since, U=0. The instantaneous surge relevant to steering is

$$DX = 0.5*X_{vv}*V^2 + 0.5*X_{DD}*D^2 + (X_{vR} + m)*V*R \quad \text{(eqn 5.4)}$$

SL/7:

MARINER:

$$DX = 0.5*X_{vv}*V^2 + 0.5*X_{DD}*D^2 + (X_{vR} + m)*V*R + (0.5*X_{RR} + m*X_{\ })*R^2 + X_{vD}*V*D + X_u*DU + 0.5*X_{uu}*DU^2 + (1/6)*X_{uuu}*DU^3 \quad \text{(eqn 5.5)}$$

From the instantaneous surge equation relevant to steering of the SL/7 Reid came up with the following cost function:

$$J = (1/2T)*\int_0^{t_o} (\text{LAMBDA}''*V*R + \text{ETA}*V^2 + D^2)*dt \quad \text{(eqn 5.6)}$$

Where $\text{LAMBDA}'' = (m + X_{vR})/0.5*X_{DD}$

$\text{ETA} = (0.5*X_{vv})/0.5*X_{DD}$

and he found that the value of the (ETA*V²) term is very small so he neglected the (ETA*V²) term, then the cost function for the SL/7 is

$$J = 0.5*\int_0^{t_o} (\text{LAMBDA}''*V*R + D^2)*dt \qquad \text{(eqn 5.7)}$$

Using the same approach for the Mariner, the following cost function was derived.

$$J = (1/2T)*\int_0^{t_o} (A1*DU + A2*DU^2 + A3*DU^3 + A4*V^2 \qquad \text{(eqn 5.8)}$$

$$- A5*R^2 + D^2 + A7*V*R - A8*V*D)*dt$$

Where $A1 = X_u/0.5*X_{DD}$ $\quad A2 = 0.5*X_{uu}/0.5*X_{DD}$

$A3 = (1/6)*X_{uuu}/0.5*X_{DD}$ $\quad A4 = 0.5*X_{vv}/0.5*X_{DD}$

$A5 = (0.5*X_{RR} + m*X_G)/0.5*X_{DD}$

$A7 = (X_{vR} + m)/0.5*X_{DD}$ $\quad A8 = X_{vD}/0.5*X_{DD}$

A5 and A8 are always negative numbers, since then A5 and A8 have a minus sign in the equation. For the calm water case and when U1=15 knots, D=2.6 degrees after 2000 seconds, values of every term in the Mariner's cost function are given below to give an idea about assumptions.

$A1*DU = -0.001939$	$A2*DU^2 = -0.00000111$
$A3*DU^3 = -0.00000000039$	$A4*V^2 = 0.000003253$
$A5*R^2 = -0.0002884$	$A7*V*R = 0.0001923$
$D^2 = 0.002059$	$A8*V*D = -0.00002609$

As seen from the above ($A2*DU^2$), ($A3*DU^3$), ($A4*V^2$) and ($A8*V*D$) terms are very small compared to others, so they

may be neglected. Also to measure the DU on shipboard is very hard although it may someday be measured by new satellite facilities, so we must not include it in the cost function. After these assumptions the cost function for the Mariner is

$$J = 0.5 * \int_0^{t_0} (-A5*R^2 + D^2 + A7*V*R)*dt \qquad \text{(eqn 5.9)}$$

The only difference between Equation (5.9) and Equation (5.7) is the ($A5*R^2$) term that is not included in the SL/7's cost function. To see the effect of the ($A5*R^2$) term on the cost function the Mariner's cost function was evaluated with and without the ($A5*R^2$) term for the calm water case and U1=15 knots, D=2.6 degree after 2000 seconds, results are:

	J
with ($A5*R^2$)	0.0025399826296
without ($A5*R^2$)	0.0022515066462

There is no big difference between these two J values, and to make the derivations similar to Reid's derivations the ($A5*R^2$) term won't be included in the Mariner's cost function but as it is known that for the Mariner the ($A5*R^2$) term is as big as the (A7*V*R) term, it would be better to consider it in the cost function. After all of the above steps the cost function of the Mariner may be written as in Equation (5.7).

V and R are hard to measure on shipboard, but (V*R) can be defined as

$$V*R = OP*\dot{YAW}^2$$

Where $R = \dot{YAW} = YAWE*w$

OP = Distance from the ship pivot point to the origin. = 0.3*L

w = Natural frequency of the ship's steering system closed loop. (w = 0.05 rad/sec was initially used.)

Finally the cost function for the Mariner is

$$J = 0.5*\int_0^{t_0}(LAMBDA*YAWE^2 + D^2)*dt \qquad \text{(eqn 5.10)}$$

Where $LAMBDA = (m + X_{\dot{VR}})*OP*w^2/0.5*X_{DD}$

Since, X depends on ship speed, for different ship speeds values of LAMBDA were calculated and are presented in Table (V). These LAMBDA values were calculated by assuming the natural frequency of the ship's steering system closed loop is equal to 0.05 radian/second. How important the accuracy of the LAMBDA value is with respect to finding the optimal control parameters will be observed in Chapter 6.

TABLE V

Values of Weighting Factor for Different Speeds of The Mariner Class Ship

ship speed (Knots)	LAMBDA
10	6.57
15	2.91
20	1.64

VI. CONTROLLER DESIGN FOR THE MARINER, USING FORTRAN PROGRAM

We coupled a function minimization subroutine to the cascade compensater that is coupled with Mariner's equations of motion and used the subroutine to adjust the controller parameters to minimize the cost function which is derived in Chapter V, and evaluate the minimum cost.The Fortran program to calculate the optimal parameters is given in Appendix C. Compensater 'A' and 'B' are used as controllers, their structures are shown in Figure (6.1).

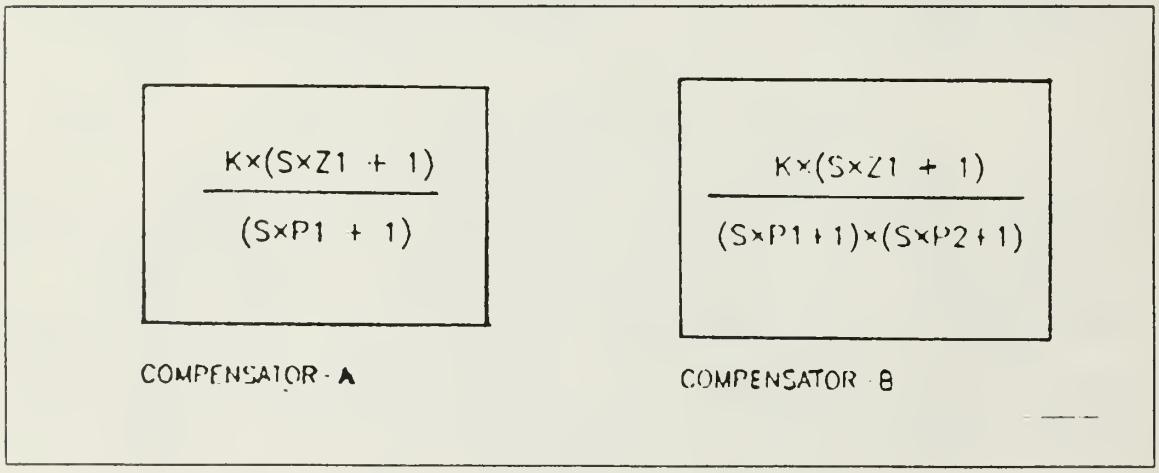


Figure 6.1 Various Structure Controllers.

A. CALM WATER CASE

For calm water, for a given 1 degree yaw command, using the computer method we optimized controllers 'A' and 'B' and the results are shown in Tables VI, VII and VIII

TABLE VI

Optimal Controller Parameters I

Simulation Results - Steady State 600 seconds

For ship speed 10 Knots the optimal parameters of various controllers and the cost

CONTR	K	Z1	P1	P2	COST
A	0.77	60.07	20.46	-----	0.0494896
B	0.74	60.07	20.12	0.902	0.0514841

TABLE VII

Optimal Controller Parameters II

Simulation Results - Steady State 600 seconds

For ship speed 15 Knots the optimal parameters of various controllers and the cost

CONTR	K	Z1	P1	P2	COST
A	0.53	51.46	18.08	-----	0.0186974
B	0.50	51.58	17.81	0.890	0.01957901

TABLE VIII

Optimal Controller Parameters III

Simulation Results - Steady State 600 seconds

For ship speed 20 Knots the optimal parameters of various controllers and the cost

CONTR	K	Z1	P1	P2	COST
A	0.40	44.87	16.06	-----	0.0094217
B	0.39	41.11	15.84	0.880	0.00991801

These results will be references for the controller design for sea state operation. We observe that increasing the speed gives us smaller controller parameters, also this behavior can be seen from the SL/7's results that are presented in [Ref. 6].

B. REGULAR SEAS CASE

The ship in regular seas is affected by sea wave disturbance forces and moments. These are functions of sea state and encounter frequency. Also the added mass and the added inertia terms are functions of sea state and encounter frequency, and the encounter frequency depends on the encounter angle and ship speed. All of these variables must be considered when calculating the optimal parameters of the controller.

Using the computer method controllers 'A' and 'B' were optimized for a few different cases and the results are shown in Tables IX, X, XI, XII, XIII, XIV, XV, XVI and XVII.

TABLE IX

Optimal Controller Parameters IV

Simulation Results - Steady State 600 seconds
optimal parameters of various controllers and the cost.
For ship speed 15 Knots, encounter angle 030.0 degree,
encounter frequency 0.50 rad./sec. and sea state 6.

CONTR	K	Z1	P1	P2	COST
A	0.358	66.6	24.61	-----	0.001252939
B	0.35	44.68	06.58	9.720	0.0010086581

TABLE X

Optimal Controller Parameters V

Simulation Results - Steady State 600 seconds
optimal parameters of various controllers and the cost.

For ship speed 15 Knots, encounter angle 060.0 degree,
encounter frequency 2.50 rad./sec. and sea state 9.

CONTR	K	Z1	P1	P2	COST
A	0.60	41.58	11.33	-----	0.000020747
B	0.70	30.49	03.37	2.940	0.000016038

TABLE XI

Optimal controller Parameters VI

Simulation Results - Steady State 600 seconds
optimal parameters of various controllers and the cost.

For ship speed 15 Knots, encounter angle 090.0 degree,
encounter frequency 0.50 rad./sec. and sea state 7.

CONTR	K	Z1	P1	P2	COST
A	0.54	30.65	09.35	-----	0.054223988
B	********** IT DID NOT CONVERGE **********				

TABLE XII

Optimal Controller Parameters VII

Simulation Results - Steady State 600 seconds
optimal parameters of various controllers and the cost.

For ship speed 15 Knots, encounter angle 120.0 degree,
encounter frequency 0.75 rad./sec. and sea state 7.

CONTR	K	Z1	P1	P2	COST
A	0.52	41.09	08.95	-----	0.018345
B	********** IT DID NOT CONVERGE **********				

TABLE XIII

Optimal Controller Parameters VIII

Simulation Results - Steady State 600 seconds optimal parameters of various controllers and the cost.

For ship speed 15 Knots, encounter angle 150.0 degree, encounter frequency 1.50 rad./sec. and sea state 8.

CONTR	K	Z1	P1	P2	COST
A	0.645	42.40	07.70	-----	0.0008188
B	********** IT DID NOT CONVERGE **********				

TABLE XIV

Optimal Controller Parameters IX

Simulation Results - Steady State 600 seconds optimal parameters of various controllers and the cost.

For ship speed 15 Knots, encounter angle 090.0 degree, encounter frequency 0.50 rad./sec. and sea state 8.

CONTR	K	Z1	P1	P2	COST
A	0.53	27.45	08.04	-----	0.070697939
B	********** IT DID NOT CONVERGE **********				

TABLE XV

Optimal Controller Parameters X

Simulation Results - Steady State 600 seconds optimal parameters of various controllers and the cost.

For ship speed 15 Knots, encounter angle 090.0 degree, encounter frequency 0.50 rad./sec. and sea state 6.

CONTR	K	Z1	P1	P2	COST
A	0.56	39.71	11.87	-----	0.019029424
B	********** IT DID NOT CONVERGE **********				

TABLE XVI

Optimal Controller Parameters XI

Simulation Results - Steady State 600 seconds
optimal parameters of various controllers and the cost.

For ship speed 10 Knots, encounter angle 060.0 degree,
encounter frequency 2.50 rad./sec. and sea state 9.

CONTR	K	Z1	P1	P2	COST
A	0.80	45.00	10.66	-----	0.000045194
B	0.89	36.27	03.22	03.22	0.000036077

TABLE XVII

Optimal Controller Parameters XII

Simulation Results - Steady State 600 seconds
optimal parameters of various controllers and the cost.

For ship speed 20 Knots, encounter angle 060.0 degree,
encounter frequency 2.50 rad./sec. and sea state 9.

CONTR	K	Z1	P1	P2	COST
A	0.47	34.67	10.35	-----	0.000014376
B	0.61	23.16	03.23	03.03	0.000011085

As seen from the above results, for all cases the compensators have characteristics of a lead network.

The effects of sea state on the controller parameters can be seen by comparing the Tables XV, XI and XIV As seen from the tables an increase in sea state causes an increase in the cost value and a decrease in the controller parameters as expected, because heavy sea state brings high disturbance forces and moments and they cause heavy yawing motions. To answer this, the controller time constants

decrease and new parameters introduce more rudder motion, so an increase in rudder motions and heading error causes increase in the cost. For ship speed 15 knots, encounter angle 90.0 degree, encounter frequency 0.5 radian/second, sea state 6 and sea state 8, rudder angles and heading errors were plotted in 200 seconds and they are shown in Figure (6.2) and (6.3).

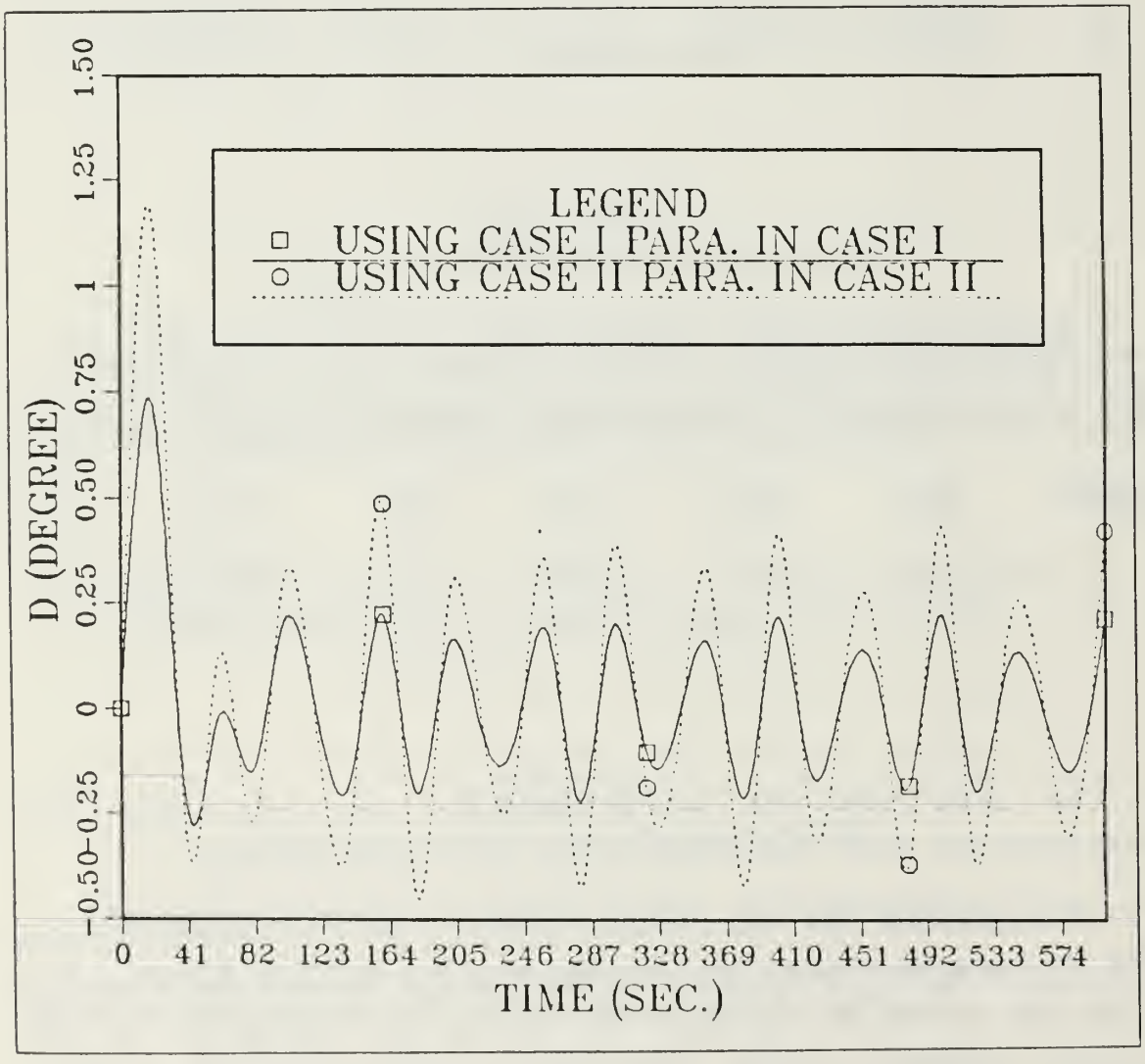


Figure 6.2 Rudder Angles in Degrees for Sea State 6 and 8.

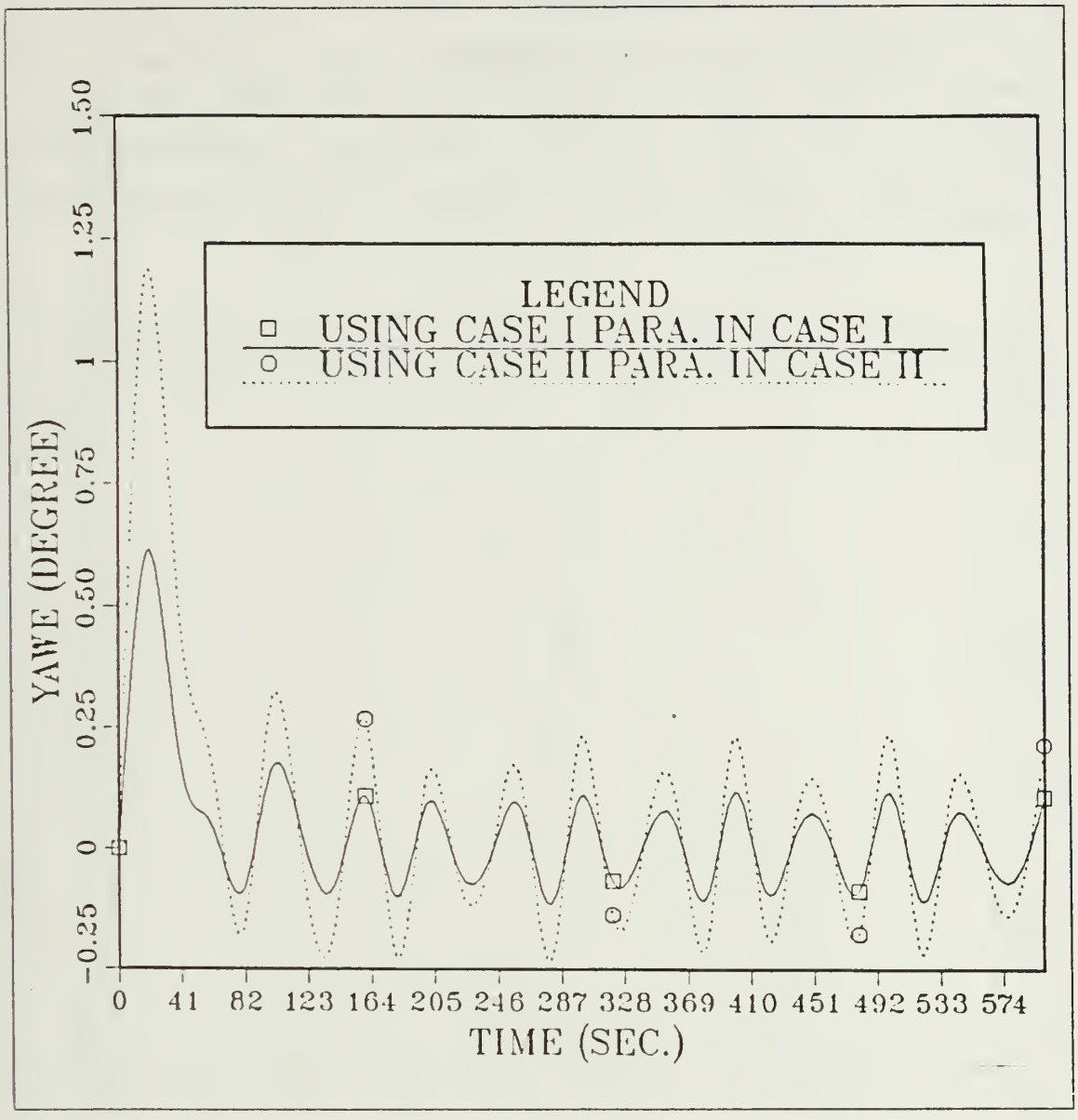


Figure 6.3 Heading Errors in Degrees for Sea State 6 and 8.

It can be seen from Figure (6.2) and (6.3), for sea state 8 rudder and heading error values are bigger than the sea state 6 rudder and heading error values, but it was noticed that the periods are almost the same for both sea states.

To have a better understanding about the effects of sea state on optimal controller parameters the cost value versus parameter values were plotted for ship speed 15 knots, encounter angle 90.0 degree, encounter frequency 0.50 radian/second and sea state 6 between 8 and they are shown in Figure (6.4), (6.5), (6.6) and (6.7).

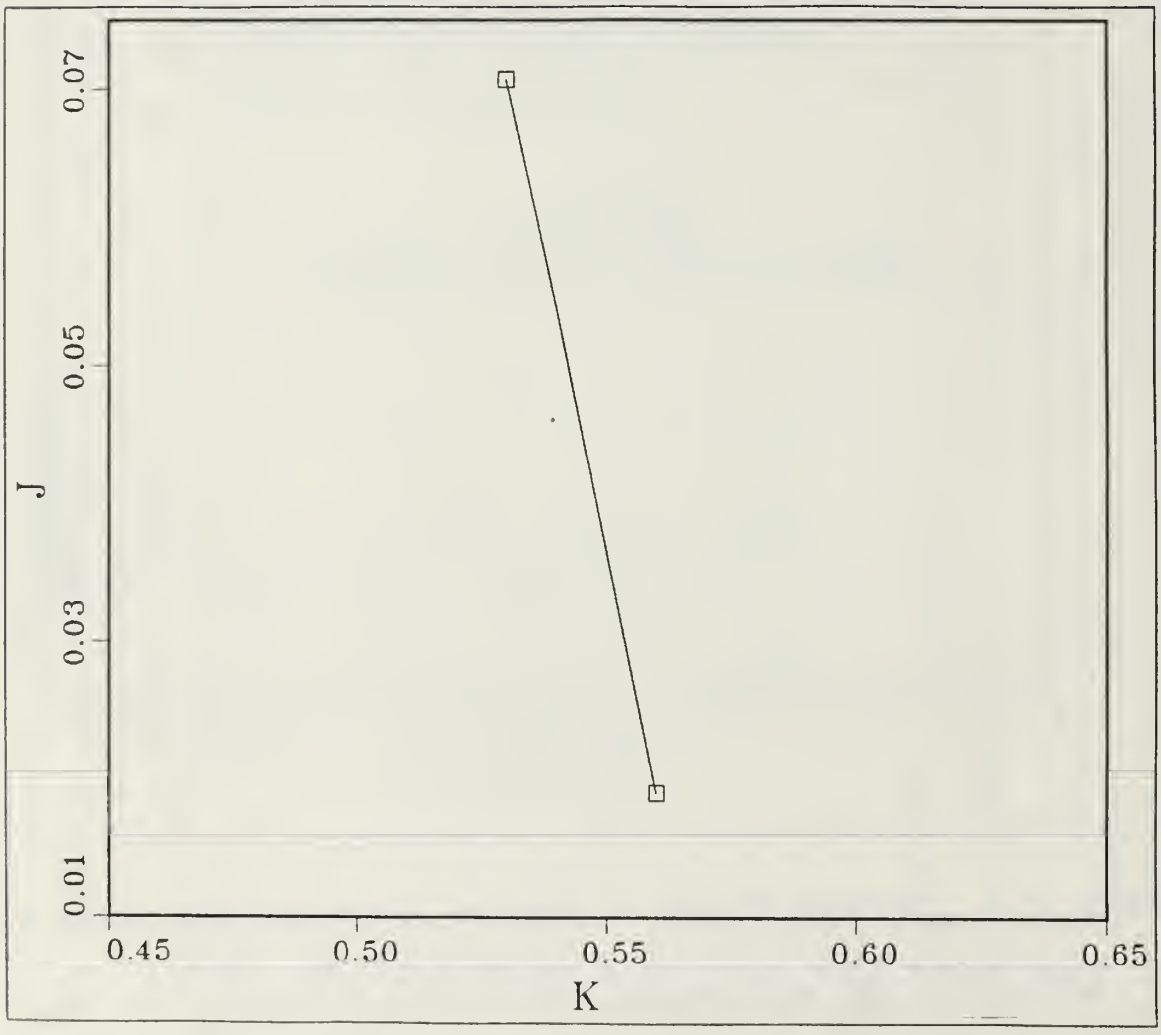


Figure 6.4 The Cost vs. K.

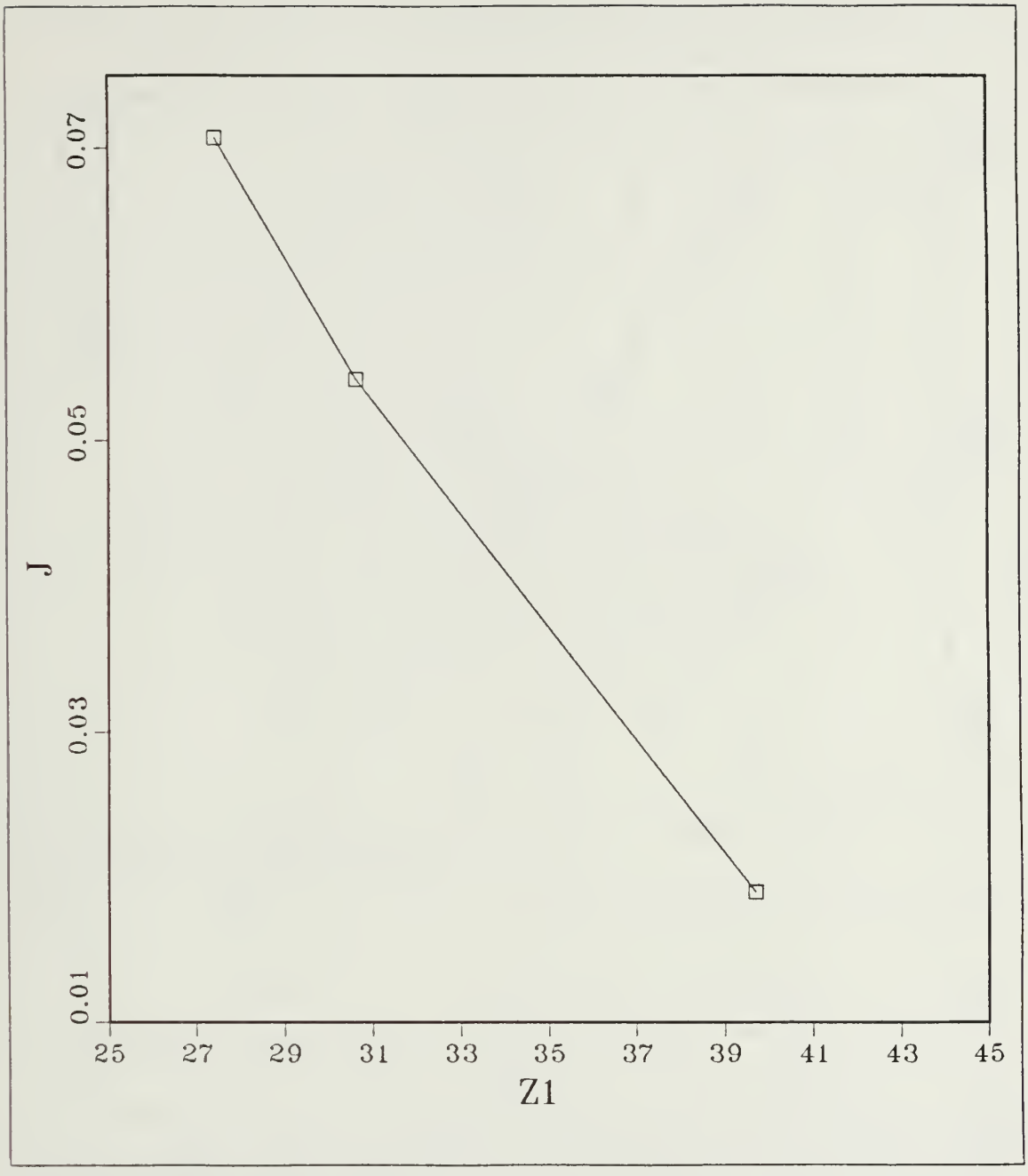


Figure 6.5 The Cost vs. Z1.

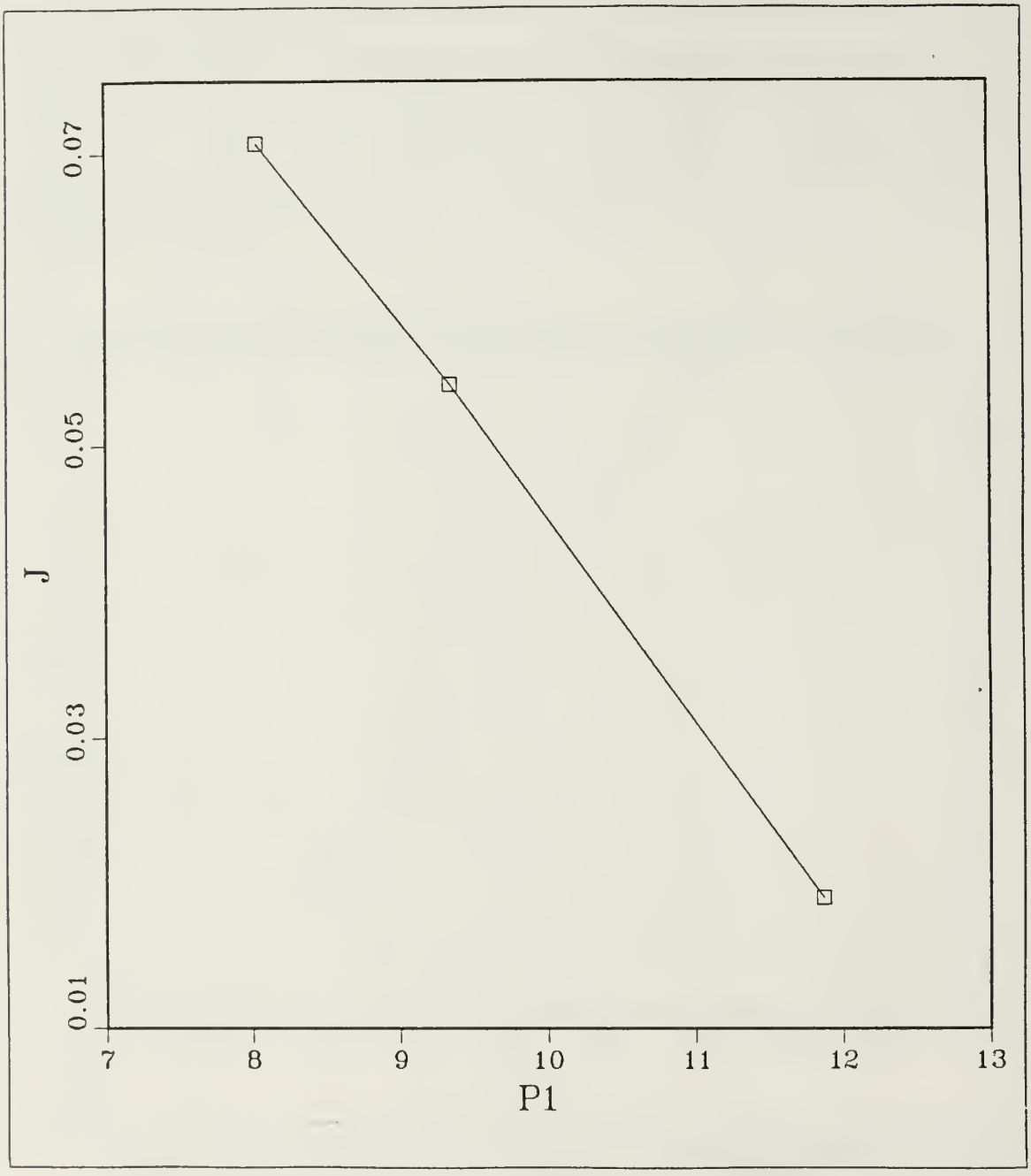


Figure 6.6 The Cost vs. P1.

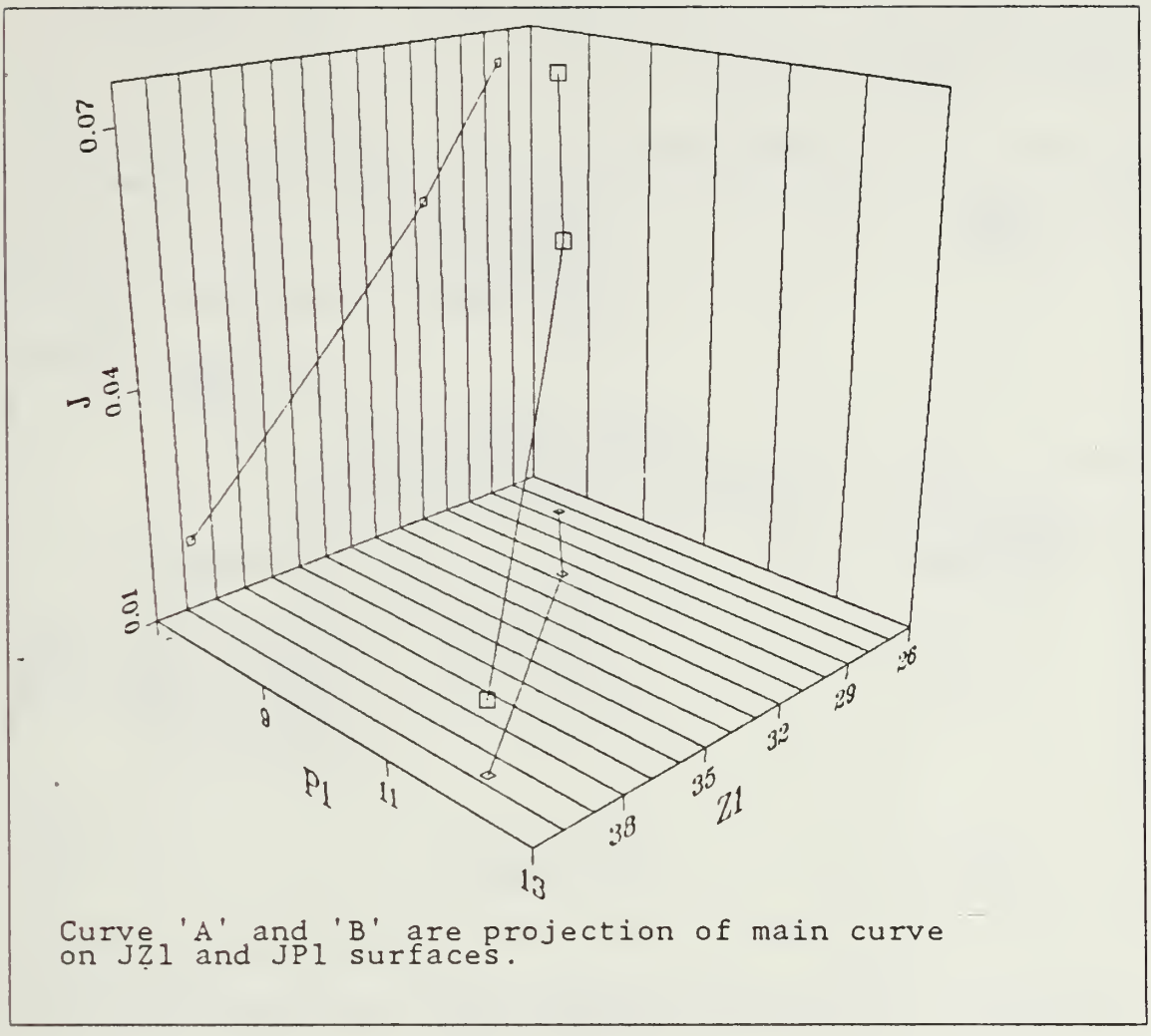


Figure 6.7 The Cost vs. Z1 and P1.

As seen from the figures the optimal parameter values have a linear behavior when changing the sea state. This property would be useful to create look up tables for parameter values. The use of look up tables will be discussed in Chapter 7.

To see the motion of the ship, the ship's equations are solved in ship coordinates (x , y) and then transformed to space coordinates (x , y), Figure (6.8) shows the orientation of space axes and moving axes.

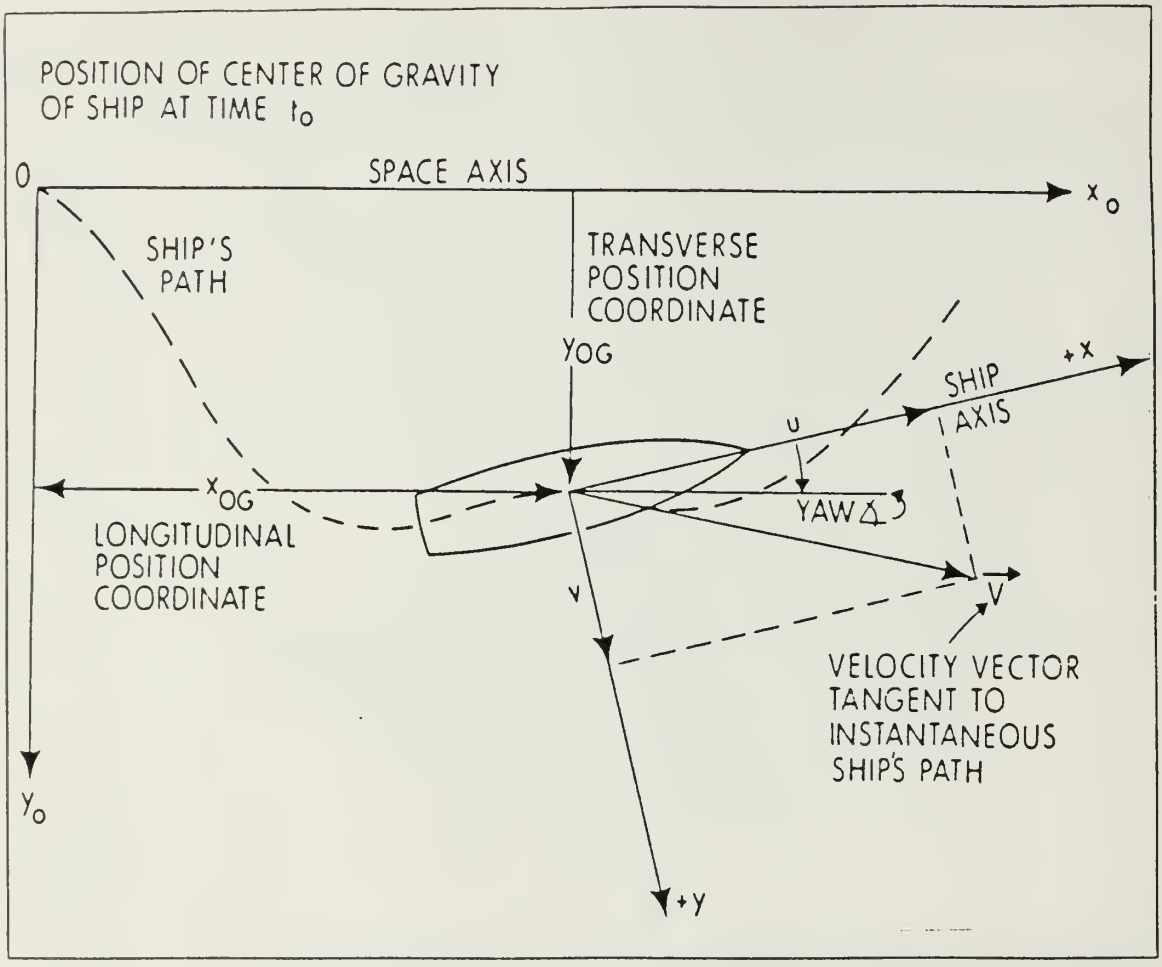


Figure 6.8 Orientation of Space Axes and Moving Axes.

The transformation from ship to space coordinates is defined as

$$\dot{X}_{og} = U*\cos(YAW) - V*\sin(YAW) \qquad \text{(eqn 6.1)}$$
$$\dot{Y}_{og} = U*\sin(YAW) - V*\cos(YAW)$$

Where Xog, Yog = Coordinates of the center of mass of the ship relative to coordinate system fixed with respect to the surface of the earth

Equation (6.1) was added into the fortran program and the ship's motion were observed for 600 seconds in calm water and in regular seas for 30.0 degree encounter angle, sea state 6, encounter frequency 0.60 radian/second and ship speed 15 knots. Both compensator 'A' and 'B' results are shown in Figure (6.9), and after 600 seconds ship's coordinates are given in Table XVIII.

TABLE XVIII
Location of the Ship

Simulation Results - Steady State 600 seconds

CASE	Xog (ft)	Yog (ft)
Calm water No rudder	15200.994873	0.00
Regular seas Comp.'A'	15199.0237114	3.2596
Regular seas Comp.'B'	15199.0763219	2.87116

For calm water and regular seas compensator 'B' gave better results than compensator 'A' did, but for some cases compensator 'B' did not converge and the difference between results is not significant. The comparisons were made as to which type of compensator brings the ship nearest to final location at the end of the 600 seconds.

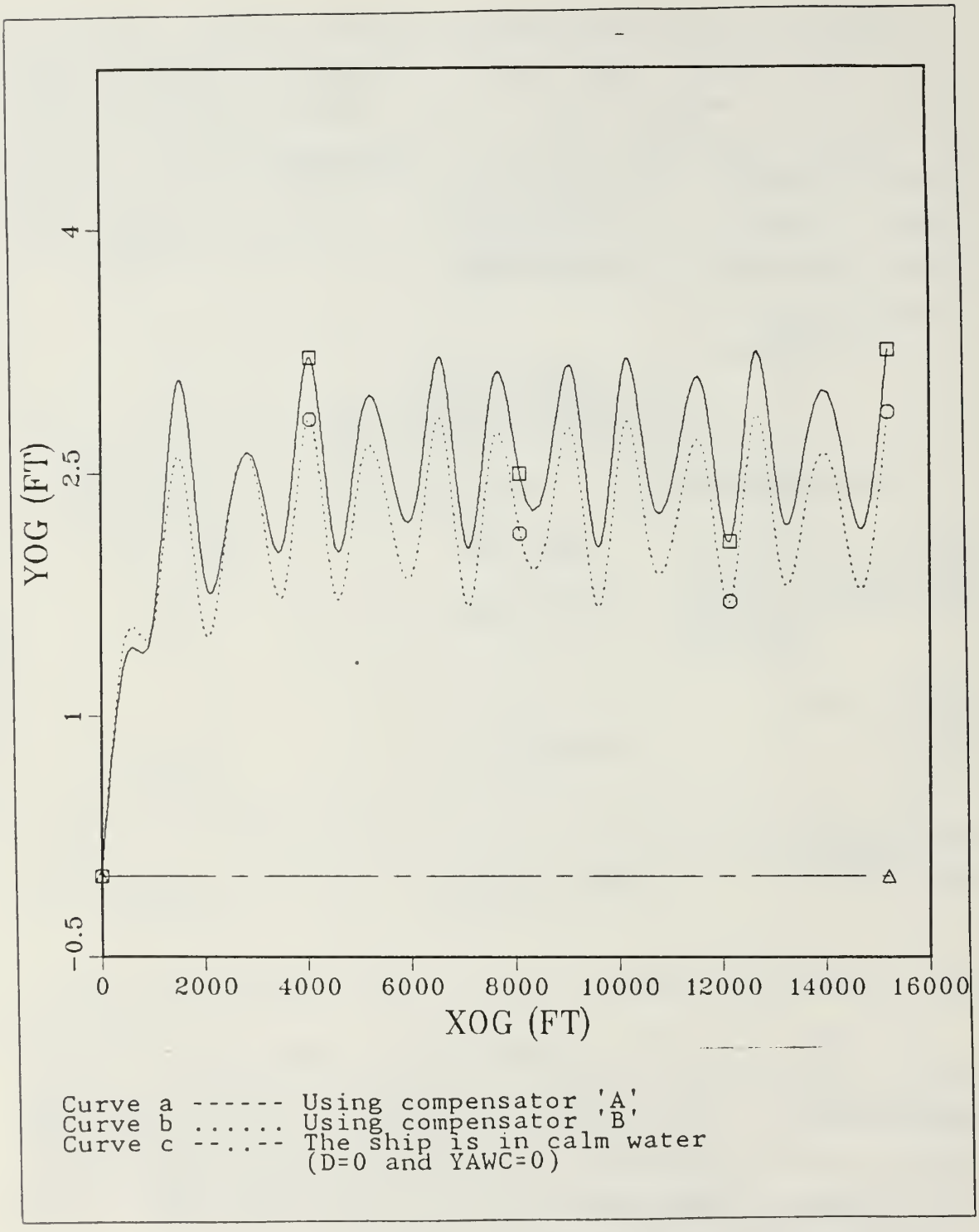


Figure 6.9 Simulation Results-Steady State 600 sec.

To see the effects of the weighting factor (LAMBDA), on the compensator parameters, different LAMBDA values, were used with compensator 'A' and the compensator parameters were computed for ship speed 15 knots, calm water and regular seas with encounter angle 90.0 degree, encounter frequency 0.5 radian/second and sea state 8. Results are shown in Table (XIX) and Table (XX).

TABLE XIX

Test for LAMBDA Value I

Simulation Results - Steady State 600 seconds
for different weighting factor values,
calm water, ship speed 15 knots,
optimal parameters of A type controller.

LAMBDA	K	Z1	P1
10.00	1.07	29.59	11.30
9.00	1.00	31.5	11.90
8.00	0.94	32.91	12.45
7.00	0.88	34.84	13.14
6.00	0.80	37.61	14.02
5.41	0.76	39.38	14.62
4.91	0.72	41.26	15.19
4.41	0.68	43.14	15.78
3.91	0.63	45.26	16.36
3.41	0.58	48.06	17.10
2.91	0.53	51.07	17.93
2.41	0.48	55.21	18.99
1.91	0.42	59.79	20.15
1.41	0.35	64.79	21.31
0.91	0.27	76.89	24.17
0.41	0.18	89.61	28.22

TABLE XX

Test for LAMBDA Value II

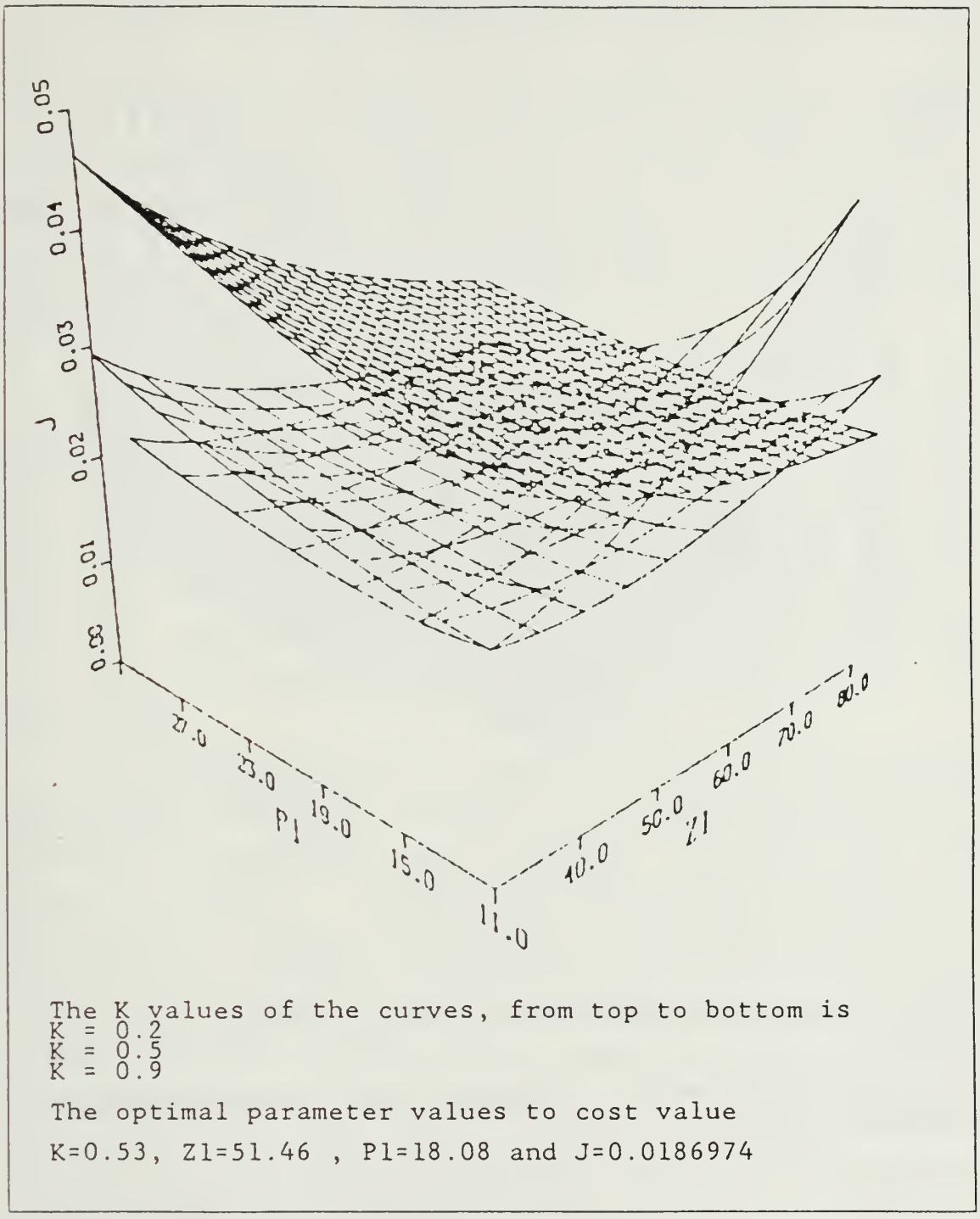
Simulation Results - Steady State 600 seconds

for different weighting factor values, regular seas, ship speed 15 knots encounter angle 90.0 degree, sea state 8, and encounter frequency 0.50 radian/second optimal parameters of A type controller.

LAMBDA	K	Z1	P1
5.41	0.71	24.91	08.41
4.91	0.68	25.36	08.37
4.41	0.64	25.88	08.32
3.91	0.60	26.34	08.22
3.41	0.56	26.85	08.11
2.91	0.53	27.45	08.04
2.41	0.48	28.28	08.04
1.91	0.44	29.16	08.16
1.41	0.40	31.21	08.35
0.91	0.34	33.95	08.71
0.41	0.26	39.00	09.27

Using above compensator values did not make a significant change on the ship location after 600 seconds simulation, so it can be said that the accuracy of the weighting factor is not very important.

A few simulation runs were performed by changing the optimal compensator parameters a small amount and the cost curve was plotted. Figure (6.10) shows the cost curves for three different K values versus Z1 and P1 when the ship is in calm water, ship speed 15 knots and 1 degree course change, and Figure (6.11) shows the cost curves for ship speed 15 knots, encounter angle 60.0 degree, encounter frequency 2.5 radian/second, and sea state 9.

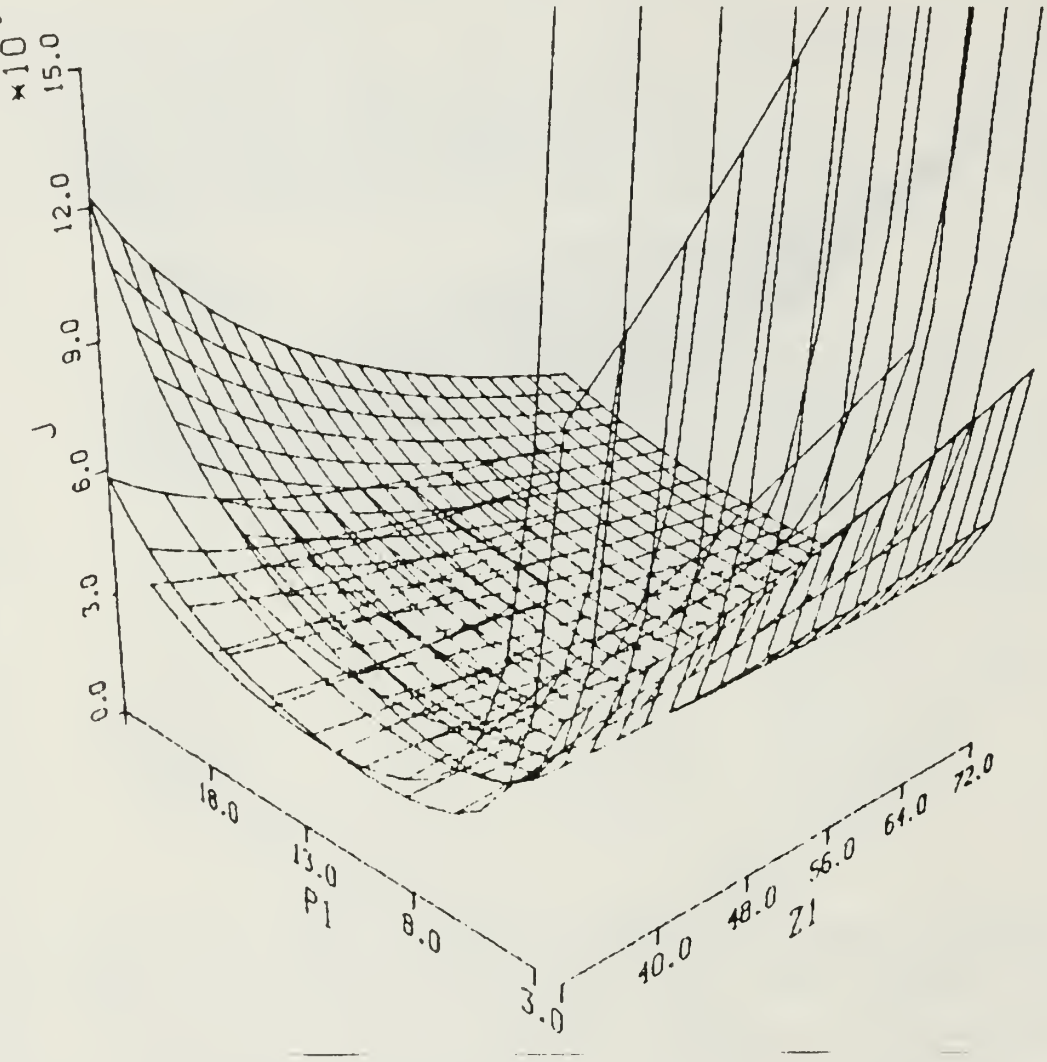


The K values of the curves, from top to bottom is
K = 0.2
K = 0.5
K = 0.9

The optimal parameter values to cost value

K=0.53, Z1=51.46 , P1=18.08 and J=0.0186974

Figure 6.10 The Cost Curves vs. Z1 and P1 when Parameters are Changing Around the Optimal Values for the Calm Water Case.



The K values of the curves, from top to bottom is
K = 0.3
K = 0.6
K = 0.9
The optimal parameter values and respect to cost value for this case:
K=0.60, Z1=41.58 , P1=11.33 and J=0.000020747

Figure 6.11 The Cost Curves vs. Z1 and P1 when Parameters Changing Around the Optimal Values for the Sea State Case.

The results showed that the cost curve is flat around the minimum. The same conclusion is made in [Ref. 7 and 8] for the SL/7. Using controller parameters on the flat portion of the cost curve, but not at the optimum did not make a significant differences in simulation results, after 600 seconds of cruising there was little change in the final location of the ship. This property of the cost surface may be useful in reducing the time required to find a pratical minimum for the cost function.

The ship speed effects on compensator parameters for regular seas can be seen by comparing Tables X, XVI and XVII. Observe that for both cases increasing the ship speed made the parameter values and the cost value decrease. The same observation can be made by comparing the results that are found for the SL/7 container ship for different speeds in [Ref. 9]. These results strongly indicate that the dynamics of the ship determines the optimum structure for the controller.

To see the stability of the system, the ship's third order Nomoto model was cascaded with the compensator and the open loop system BODE plot was drawn, in every case the system is stable. The phase margin varies between 40 degrees and 70 degrees and the zero cross over of the magnitude curve is around 0.04 radian/second. It was observed that small changes in the compensator parameters do not affect stability. With in large limits of parameter variation the system is always stable. For regular seas, ship speed 15 knots, encounter angle 30.0 degrees, encounter frequency 0.6 radian/second and sea state 6, using compensator 'A' and 'B', the structure of the system is presented in Figure (6.12) and the system open loop BODE plot and NICHOLS plot are shown in Figures (6.13) and (6.14).

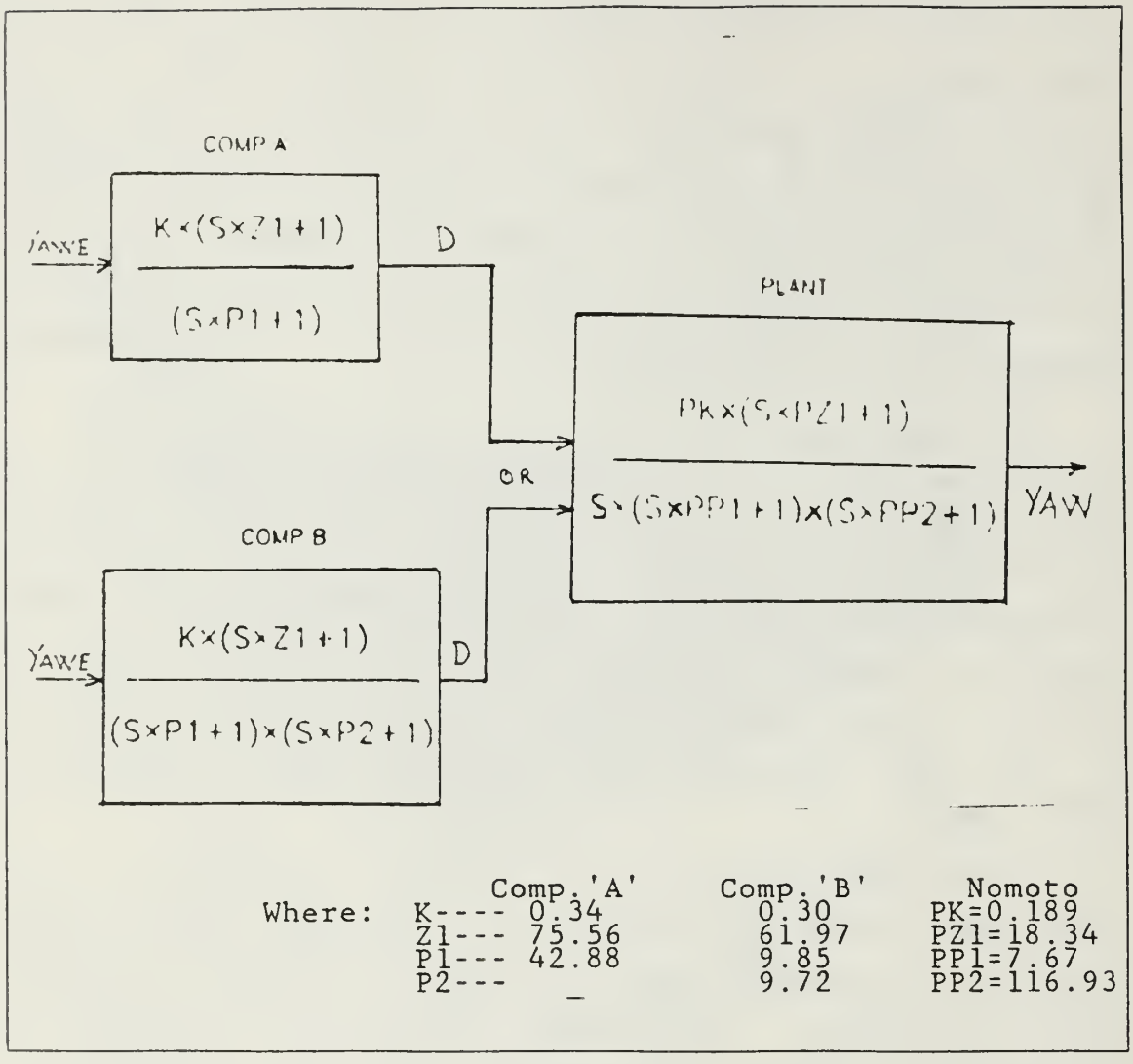


Figure 6.12 Open Loop Steering Model.

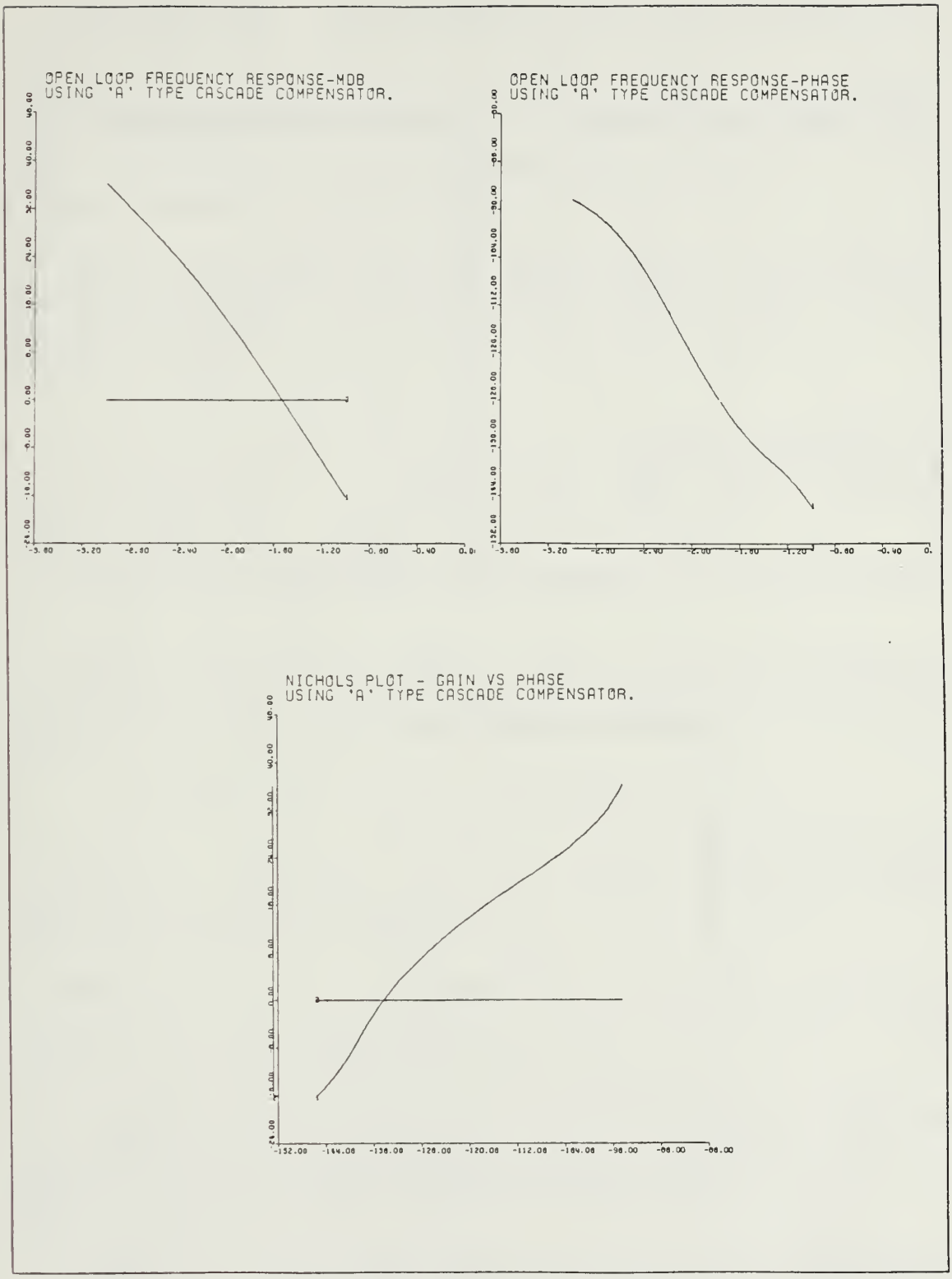


Figure 6.13 Open Loop System BODE and NICHOLS Plots. (Using Comp.'A').

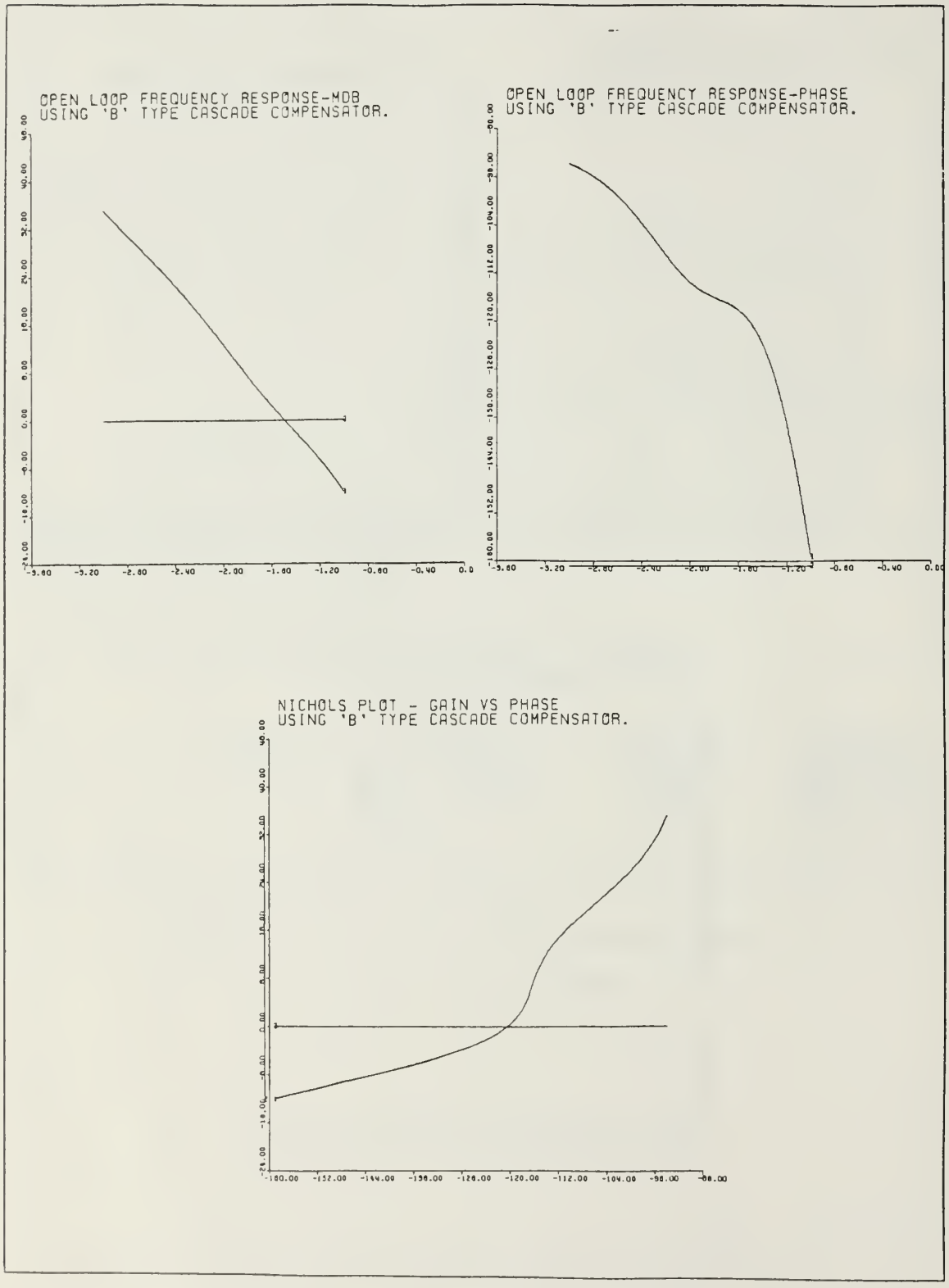


Figure 6.14 Open Loop System BODE and NICHOLS Plots. (Using Comp. 'B').

Changing the environmentel conditions changes the optimal controller parameters. A few simulations were run changing the environmental conditions only but the optimal controller parameters were kept unchanged The behavior of rudder and heading error were observed. Three cases were defined depending on the enviromental conditions.

Case I :Encounter angle 90 degree, encounter frequency 0.5 radian/second, sea state 6 and ship speed 15 knots. Optimal controller parameters for Case I are

K = 0.56 Z1 = 39.71 P1 = 11.87

Case II :Encounter angle 90 degree, encounter frequency 0.5 radian/second, sea state 8 and ship speed 15 knots. Optimal controller parameters for Case II are

K = 0.53 Z1 = 27.45 P1 = 08.04

Case III:Encounter angle 30 degree,encounter frequency 0.5 radian/second, sea state 6 and ship speed 15 knots. Optimal controller parameters for Case III are

K = 0.358 Z1 = 66.60 P1 = 24.61

Figure (6.15) and figure (6.16) show rudder and heading error when the ship is in Case I condition using Case I and Case II parameters.

Figure (6.17) and figure (6.18) show rudder and heading error when the ship is in Case II condition using Case II and Case I parameters.

Figure (6.19) and figure (6.20) show rudder and heading error when the ship is in Case III condition using Case III and Case I parameters.

Figure (6.21) and figure (6.22) show rudder and heading error when the ship is in Case I condition using Case I and Case III parameters.

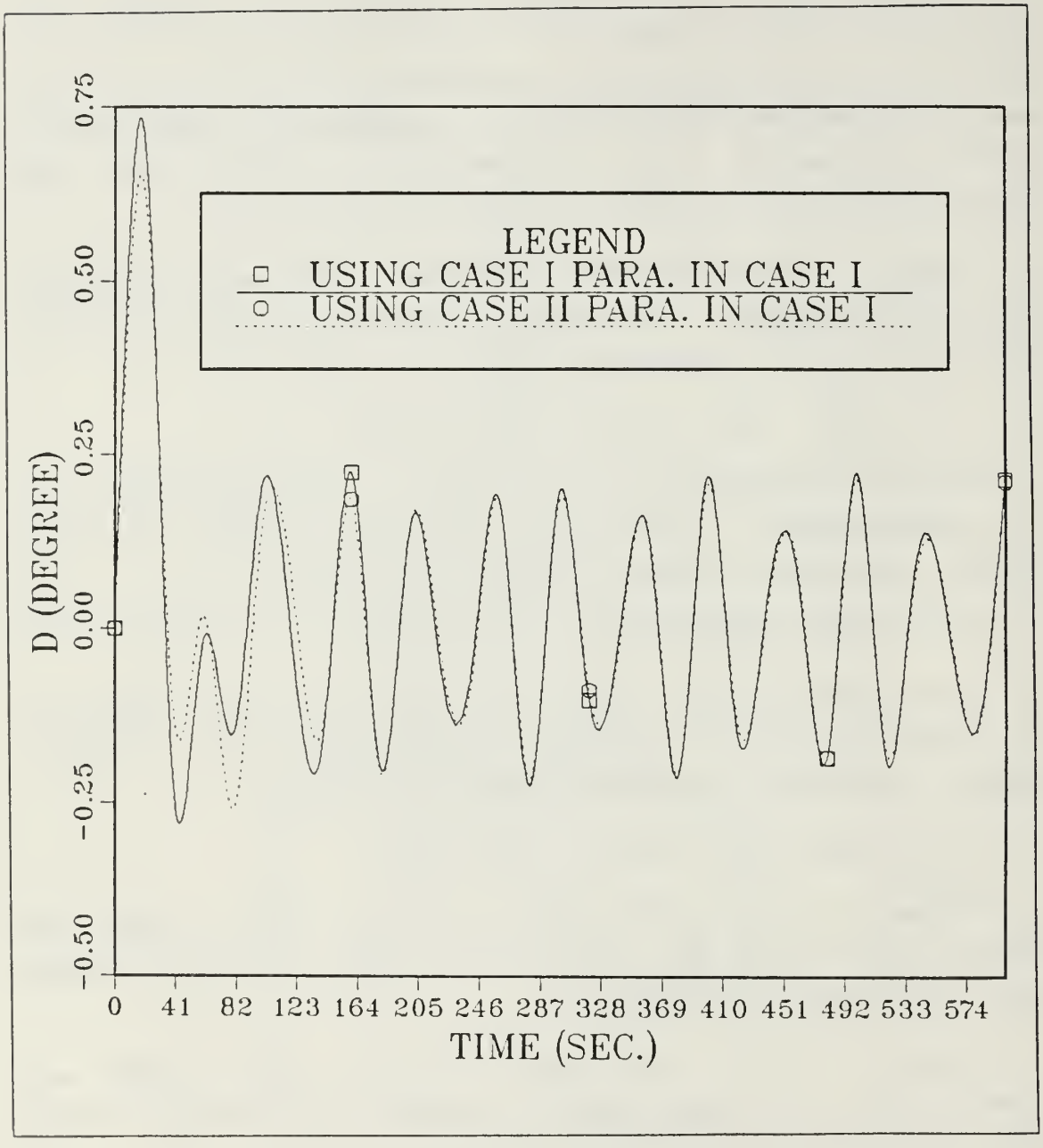


Figure 6.15 Rudder Motion in Case I with Case I and Case II Parameters.

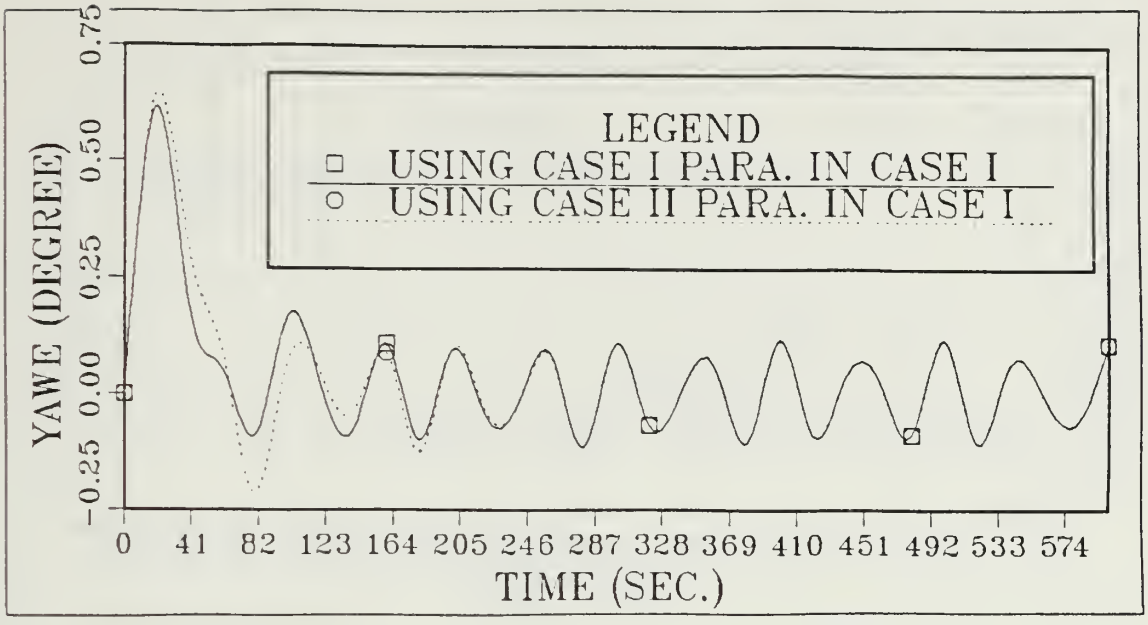


Figure 6.16 Heading Error in Case I with Case I and Case II Parameters.

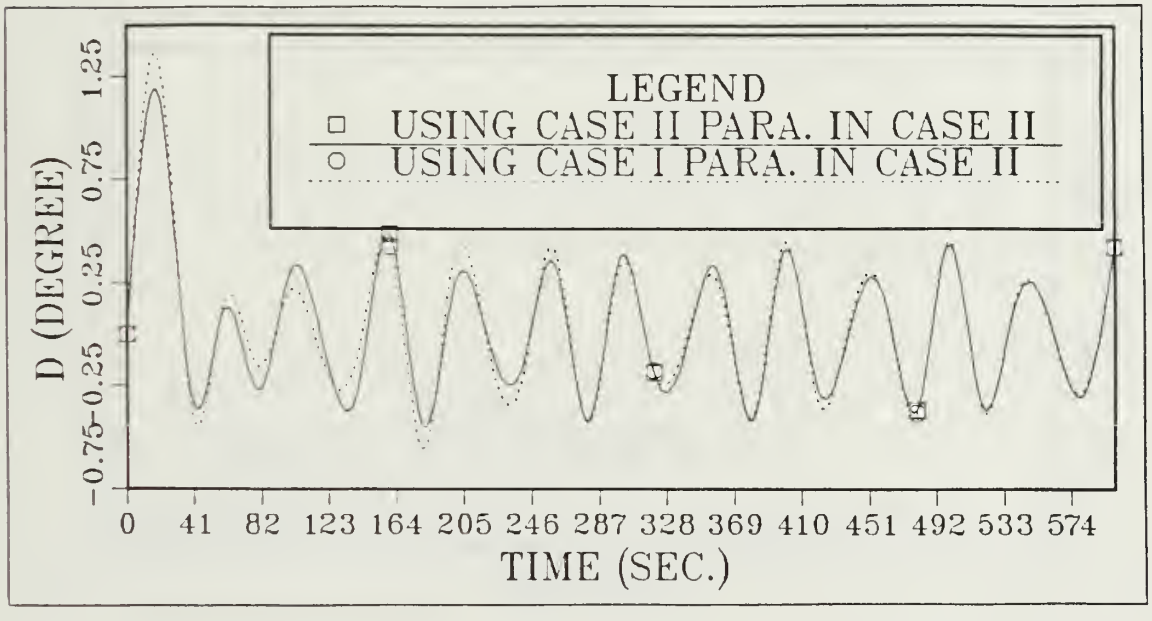


Figure 6.17 Rudder Motion in Case II with Case II and Case I Parameters.

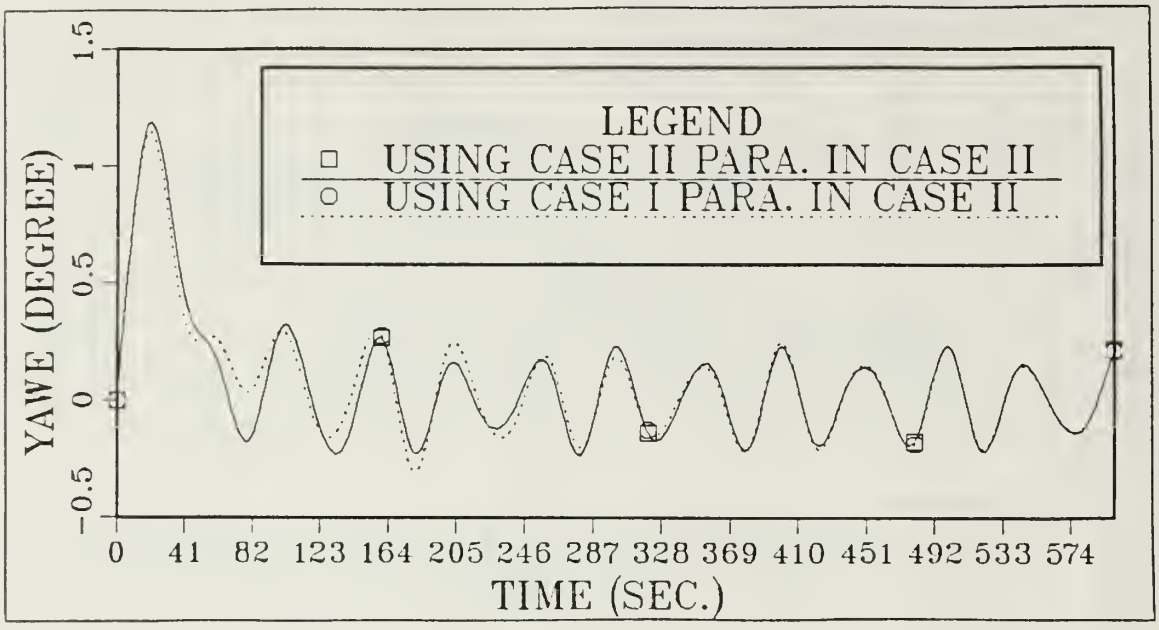


Figure 6.18 Heading Error in Case II with Case II and Case I Parameters.

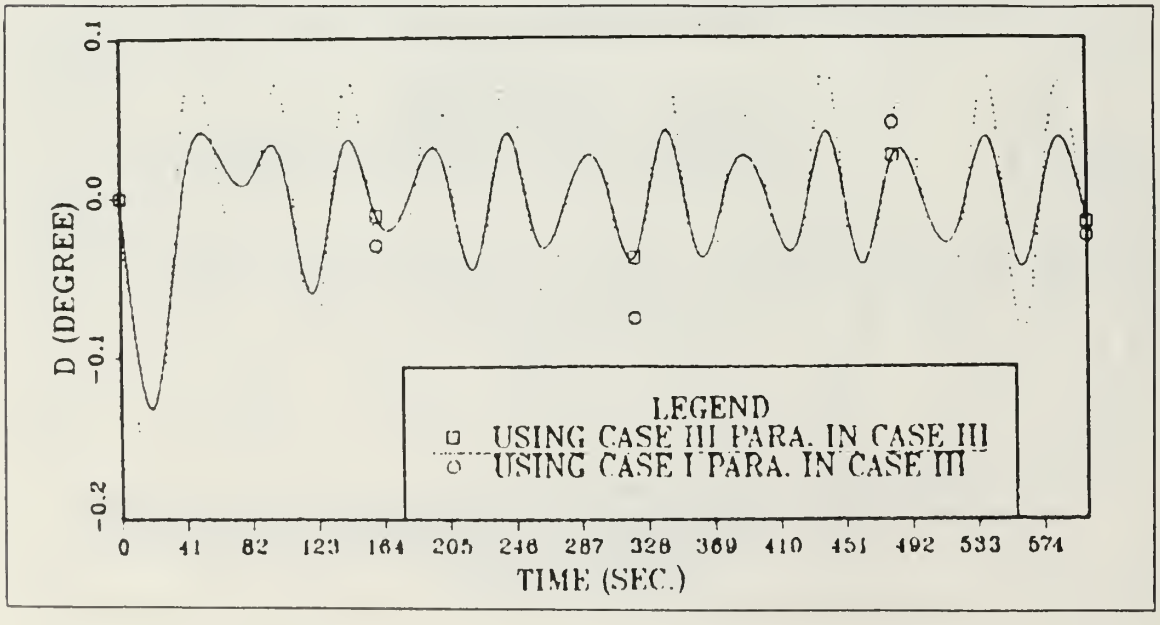


Figure 6.19 Rudder Motion in Case III with Case III and Case I Parameters.

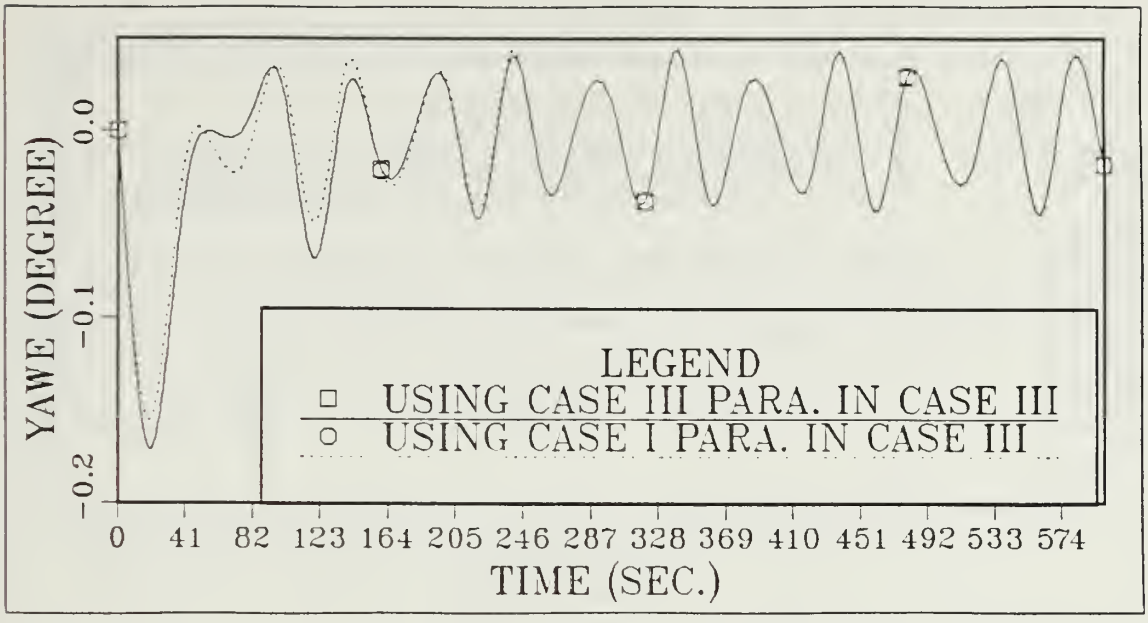


Figure 6.20 Heading Error in Case III with Case III and Case I Parameters.

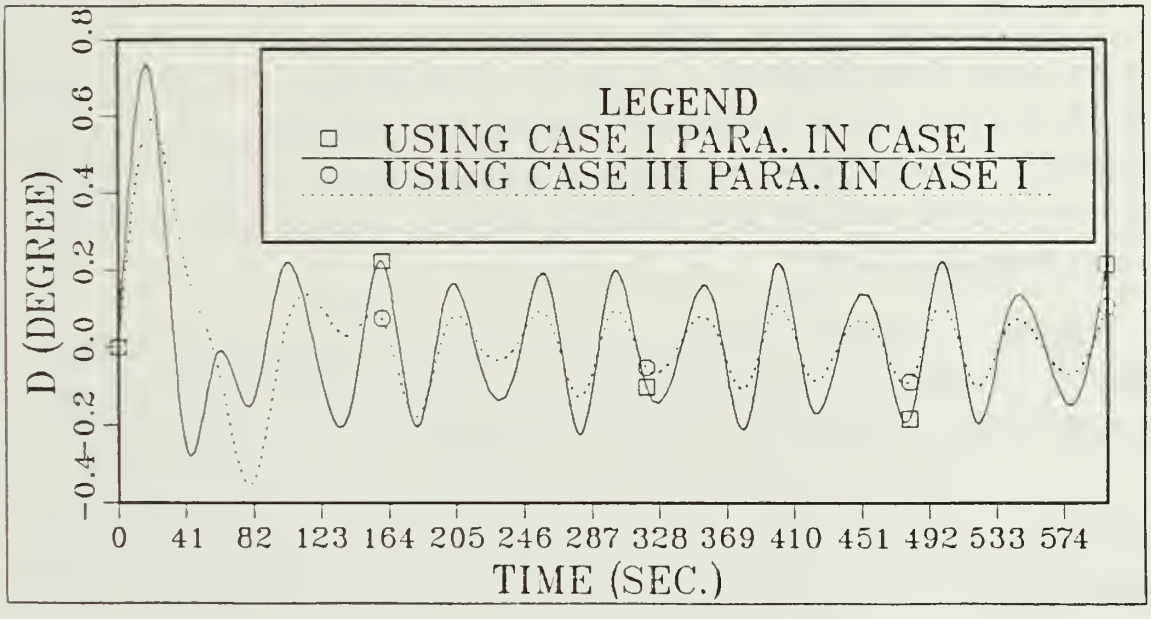


Figure 6.21 Rudder Motion in Case I with Case I and Case III Parameters.

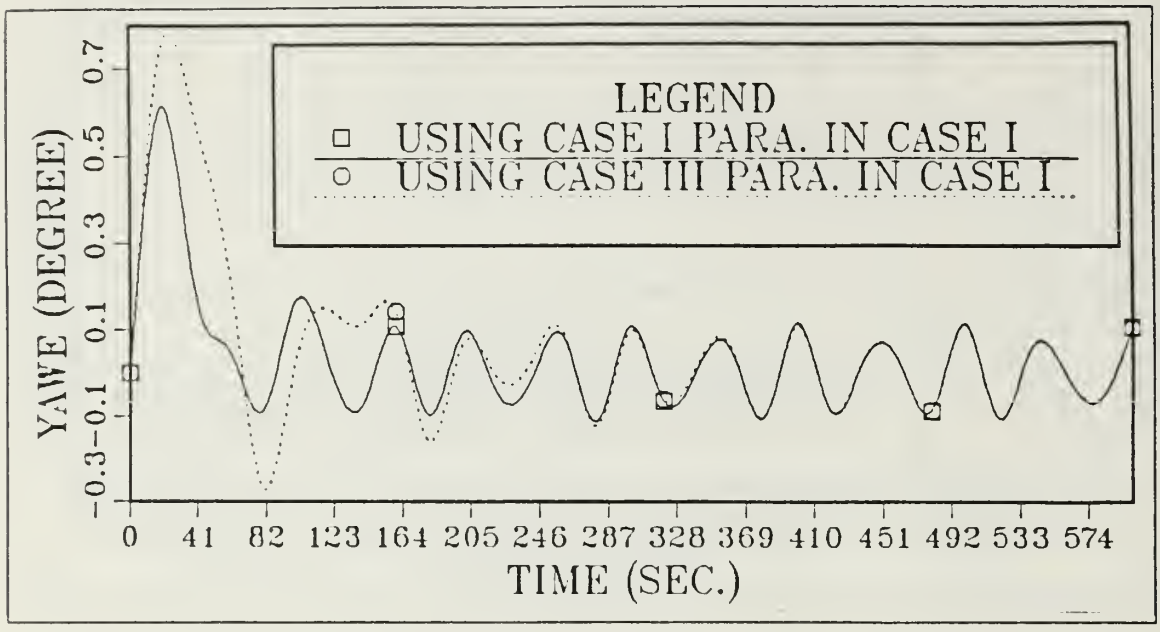


Figure 6.22 Heading Error in Case I with Case I and Case III Parameters.

As seen from Figures (6.15), (6.16), (6.17) and (6.18) using Case I optimal parameters in Case II or Case II optimal parameters in Case I did not make a big difference in rudder motion and heading error, except in the transient response part.

Figures (6.19) and (6.20) show that to use Case I optimal parameters when ship is in Case III is not proper because those parameters increase rudder and heading error.

As seen from Figure (6.21), using Case III parameters in Case I provided better rudder motion after the transient response part. Cost values were calculated and it was observed that when the ship is in Case I using Case I optimum parameters it gave a better cost value than Case III parameters did, end of the 600 seconds simulation. But it

was also observed that if the final time of simulation is extended, the difference between the cost values decreases and if we continue to increase the final time, using Case III parameters in Case I gives a smaller cost value than Case I parameters do.

These results show that the transient response part is important in finding the optimum control parameters.

VII. AN APPROACH TO AN ADAPTIVE AUTOPILOT

As seen from the previous chapter the optimal controller parameters change for changes in ship conditions such as ship speed, loading etc. and also changes in environmental conditions such as sea state, encounter angle encounter frequency and depth of water. To maintain optimal steering performance automatically in the presence of changing conditions, it is necessary to design an adaptive system which is capable of self adjustment of the controller parameters to provide minimum added drag due to steering.

The steering control system, if desired as an adaptive autopilot, would consist of four Subsystems as shown in Figure (7.1).

- Subsystem #1 would be a computer which will perform to find the optimal control parameters. It should get the information about ship steering characteristics from system state sensors such as a gyrocompass, rudder angle potentiometer and speed log and it should feed the control parameter values to the controller.
- Subsystem #2 would be the controller, which should be adjustable. It gets the parameter values from subsystem #1 and sends the rudder command to subsystem #3 which steers the ship.
- Subsystem #3 is the plant which includes the system state sensors and ship steering devices such as servo motors, hydraulic pumps, rudder, steering gear etc.
- Subsystem #4 is a manual control option for safety rules. If manual steering is needed it cuts the connection between controller and steering devices and gives the control to the helmsman.In case of computer

failure it cuts the connection between computer and controller and sends to the controller parameter values which are chosen by the watch officer. It contains look up tables which specify control parameters as functions of steering and environmental conditions.

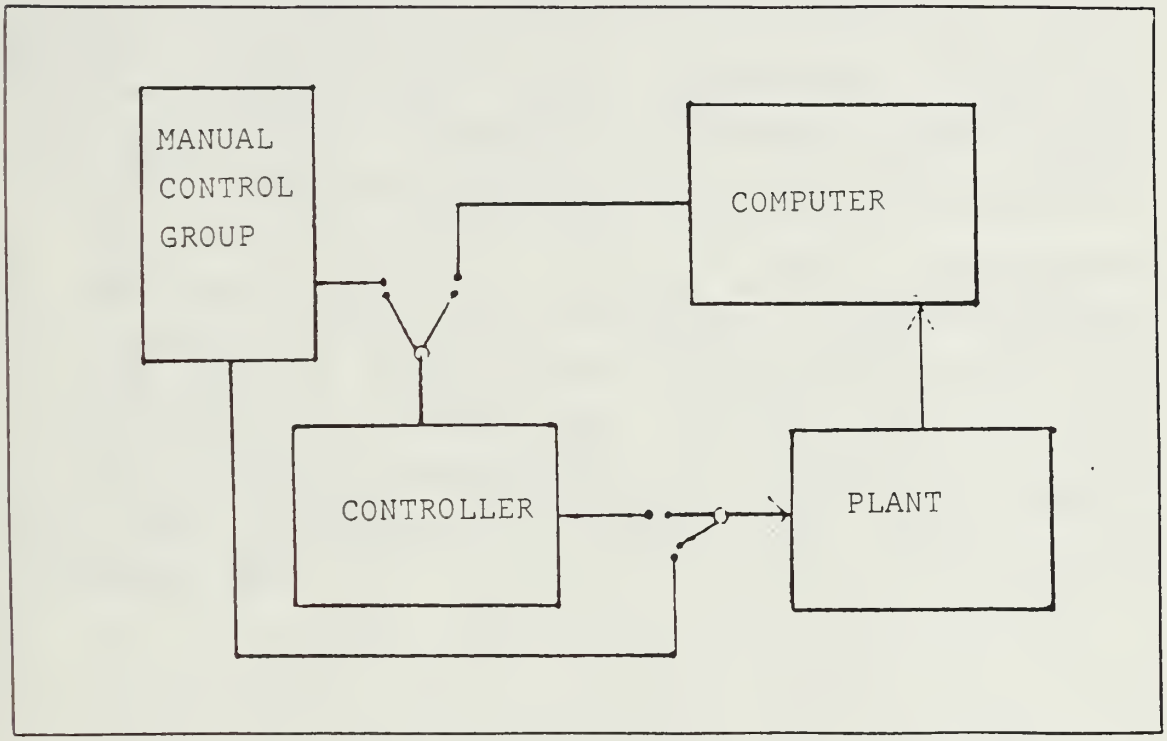


Figure 7.1 Adaptive Control Scheme.

Success of the this adaptive autopilot system depends on the computer program as well as the accuracy of the system sensors. The computer program which was used in this research may be used on board, but the present program minimization subroutine needs a lot of computation time and it also needs starting guesses for parameters which are different for every condition. If computation time is reduced to a reasonable time and starting values are made

available in proper limits, the modified function minimization program would be considered as an adequate program for on board purposes. The work on reducing the computation time is presented in [Ref. 7].

In the future, ships could have better measurement of navigation than can be provided by conventional equipment on board. For example, the U.S Navy is involved in a program to build a system which is called NAVSTAR/GLOBAL POSITION SYSTEM (GPS). The system will provide extremely accurate three-dimensional position and velocity information to users anywhere in the world. And also another system is called NAVY REMOTE OCEAN SENSING SYSTEM (NROSS) will be able to determine wind velocities over th world's oceans with an accuracy sufficient to determine ocean surface waves. Using such valid information the watch officer can use look up tables and insert them into the computer, so system operation will be very close to the minimum cost value and the function minimization program will accomplish the fine tuning rapidly. Detailed information about GPS and NROSS can be found in [Ref. 10, 11, 12, 13].

VIII. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE STUDY

A. CONCLUSIONS

The conclusions resulting from this research of the Mariner class ship fuel consumption as it relate to steering might be listed as follows.

- A steering control system would minimize propulsion losses due to steering and maintain desired heading with reasonable heading error in every ship and enviromental condition.
- The study shows that the best model to describe the dynamics of the ship is the Taylor's series expansion, which allows both linear and nonlinear terms in the ship's equation of motions. Also the third order ship Nomoto model is reasonable to use instead of the ship's equation of motions. It involves both the sway and yaw equations.
- It is believed that the cost function which is presented in Chapter 6 and is commonly used by many researchers is an adequate function for on board use. It has variables such as heading error and rudder angle which can be easily measured on board, and a weighting factor (LAMBDA) which is also easy to calculate but depends on ship conditions such as ship speed. Results in chapter 7 show that accuracy of the LAMBDA value does not make significant changes in the controller.
- In this study two different types of controller were tried which have been called controller 'A' and 'B'. The structure of these controllers is shown in Chapter 7. Controller 'A' was determined to be a best structure, because in some cases the adaptive calculations for controller 'B' did not converge.

- An adaptive controller which minimizes propulsion losses due to steering is needed when ship characteristics and environmental conditions change.
- For performance in fuel saving, an adaptive controller may be better than the existing Universal Gyropilot (UGP).
- Since, equations of motion of surface ships differ only in the numerical values of the coefficients, the simulation programs used for the Mariner class ship would be useable for other type of ships, knowing their hull characteristics. The studies made for the SL/7 containership are examples. [Ref. 6, 9, 7, 8].

B. RECOMMENDATIONS FOR FUTURE STUDY

This research does not cover all ship and enviromentel conditions, so to get a better understanding about optimal controller parameters, it is recommended that the methods be applied to find the controller parameters for an expanded range of operating conditions.

This thesis investigated only course_keeping with emphasis on minimizing rudder and yawing activity to reduce fuel consumption. If a track following control or an automatic control for replenishment at sea is desired, a different cost function might be needed. The nature of the cost function for such applications should be studied.

Irregular seas case were not considered in this study. For future work it is necessary to study the effect of irregular seas on the controller parameters and on the cost function. It is believed that compering the regular sea results with irregular sea results would give better understanding about ship's steering characteristics.

The transient response has a big affect on finding the optimum parameter values. For future studies it may be better to increase final time so that the effect of the transient response would not be significant.

Recent studies on roll stabilization shows that using rudder stabilization is successful in reducing roll, [Ref. 14, 15]. Adding the roll equation into the ship model and determining a proper cost function for minimum roll motion a controller could be designed for roll stabilization.

APPENDIX A

THESIS FORTRAN

```
$JOB
      REAL*8 L,L2,L3,L4,L5,L6
      REAL*8 X,XDOT,Y,YDOT,U,UDOT,V,VDOT,YAW,R,RDOT
      REAL*8 TIME,ETIME,X1,X2,X3,X4,X5,X6,X7,X8
      REAL*8 Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8
      REAL*8 N0,N1,N2,N3,N4,N5,N6,N7,N8
      REAL*8 C1,C2,C3,C4,C5,F1,F2,F3
      REAL*8 RO,DELT,S,DU,U1,K,Z1,Z2,P1,P2
      REAL*8 DYAWE,YAWE,YAWC,ISR,ISE,TDIFF,LAMDA
      REAL*8 S1,S2,DS1,DS2,D
      REAL*8 MASS,IZ,XG,YVDOT,NVDOT
      REAL*8 YR,YRDOT,NR,NRDOT,FX,FY,MZ
      REAL*8 RX,RY,RZ,TX,TY,TZ,WA,WE
      REAL*8 RXR,RYR,RXI,RYI,MZR,MZI
C  INITIAL CONDITIONS FOR INTEGRATION
C  SIMULATION END TIME IN SECONDS
      ETIME=210.
      TIME=0.0
      ICOUNT=1
C  INITIALIZE THE COST FUNCTION
      ISE=0.0
      ISR=0.0
      TDIFF=0.0
      LAMDA=2.91
C  GAIN COEFFICIENTS TO BE OPTIMIZED
      K=0.5
      Z1=43.32
      P1=9.26
C  X,XDOT,Y,YDOT ARE FIX COORDINATES ON EARTH
      X=0.0
      Y=0.0
      XDOT=0.0
      YDOT=0.0
C  U,UDOT,V,VDOT ARE FIX COORDINATES ON SHIP
      F1=0.0
      F2=0.0
      F3=0.0
      V=0.0
      UDOT=0.0
      VDOT=0.0
      YAW=0.0
      YAWE=0.0
      R=0.0
      RDOT=0.0
C  ORDERED SPEED IN  FEET/SEC
C   15.*1.689 FT/SEC=15 KNOTS
      U1=15.*1.689
C  AT STEADY STATE ACTUAL SPEED (U) = COMMAND SPEED (UC)
      U=U1
C  D =  RUDDER ANGLE
      D=0.0/57.296
      L=520.
      L2=L**2
      L3=L*L*L
      L4=L*L3
      L5=L*L4
      L6=L*L5
C     FORCES IN X,Y DIRECTION COMPUTED IN FORCES
C     MOMENTS IN Z
```

```
      FX=0.
      FY=0.
      MZ=0.
      RXR=-.37126D5
      RXI=.68406D5
      RYR=-.39983D6
      RYI=.2447D6
      MZR=.296D8
      MZI=-.17546D8
      RX=(RXR**2+RXI**2)**.5
      RY=(RYR**2+RYI**2)**.5
      RZ=(MZR**2+MZI**2)**.5
      TX=DATAN2(RXI,RXR)
      TY=DATAN2(RYI,RYI)
      TZ=DATAN2(MZI,MZR)
C     SIGNIFICANT WAVE HEIGHT:(SEA STATE 2)
      WA=25.
C     ENCOUNTER FREQUENCY:(WHEN ENCOUNTER ANGLE IS 00)
      WE=0.75
C     ADDED MASS AND ADDED INERTIA TERMS:
      MASS=.14685D+07
      IZ=.23567D+11
      XG=-23.9
      YR=-.95066D8
      YRDOT=.12211D8
      NR=-.13152D11
      NRDOT=-.72177D10
      YVDOT=-.72459D6
      NVDOT=-.53846D8
C
C HYDRODYNAMIC COEFFICIENTS ARE INSERTED HERE AS PARAMETERS
      RO=1.9876*.5
      YAWE=0.0
      DS1=0.0
      DS2=0.0
      S1=0.0
      S2=0.0
  200 CONTINUE
C  INPUT YAW COMMAND
      YAWC=0.0
C     IF (TIME.GE.0.0) YAWC=(1.0/57.296)
C  ERROR SIGNAL TO DRIVE RUDDER(YAW ACTUAL - YAW ORDERED)
      YAWE=YAW - YAWC
      S=((U*U)+(V*V))**.5
      DU=U-U1
      DS1=(YAWE - S1)/P1
      D=(S1 + DS1*Z1)*K
   AXIAL FORCE HYDRODYNAMIC COEFFICIENTS (SURGE)
      X1=(-0.0253)*(RO*L2*S)
      X2=(0.00948)*(RO*L2)
      X3=(-0.00217)*(RO*L2/S)
      X4=(-0.189)*(RO*L2)
      X5=(0.00379)*(RO*L4)
      X6=(-0.02)*(RO*L2*S*S)
      X7=(0.168)*(RO*L3)
      X8=(0.0196)*(RO*L2*S)
C  LATERAL FORCE HYDRODYNAMIC COEFFICIENTS (SWAY)
C     Y0=(-0.0008)*(RO*L2*S*S)
      Y0=0.0
      Y1=(-0.244)*(RO*L2*S)
      Y2=(-1.702)*(RO*L2/S)
      Y3=(-0.0008)*(RO*L4*S*S)
C     Y4=(-0.105)*(RO*L3*S)
      Y4=(YR-MASS)
      Y5=(3.23)*(RO*L3/S)
      Y6=(0.0586)*(RO*L2*S*S)
      Y7=(-0.00975)*(RO*L2*S*S)
      Y8=(0.25)*(RO*L2)
C  MOMENT ABOUT Z-AXIS HYDRODYNAMIC COEFFICIENTS (YAW)
```

```
C      N0=(0.00059)*(RO*L3*S*S)
       N0=0.0
       N1=(-0.0555)*(RO*L3*S)
       N2=(0.345)*(RO*L3/S)
       N3=(0.00264)*(RO*L3*S)
C      N4=(-0.0349)*(RO*L4*S)
       N4=(NR-MASS*XG*S)
       N5=(-1.158)*(RO*L4/S)
       N6=(-0.0293)*(RO*L3*S*S)
       N7=(0.00482)*(RO*L3*S*S)
       N8=(0.1032)*(RO*L3)
C   COMMON COEFFICIENTS
       C1=(0.177)*(RO*L3)
c      C2=(0.327)*(RO*L3)
       C2=(MASS-YVDOT)
C      C3=(-0.0018)*(RO*L4)
       C3=(MASS*XG-YRDOT)
c      C4=(0.0175)*(RO*L5)
       C4=(IZ-NRDOT)
C      C5=(-0.00478)*(RO*L4)
       C5=(MASS*XG-NVDOT)
C      REGULAR WAVES
       FX=WA*RX*DCOS(WE*TIME+TX)
       FY=WA*RY*DCOS(WE*TIME+TY)
       MZ=WA*RZ*DCOS(WE*TIME+TZ)
C
       IF(TIME.EQ.0.0) FX=0.0
       IF(DABS(FY).LT.0.00000001) FY=0.0
       IF(DABS(MZ).LT.0.00000001) MZ=0.0
C   EQUATIONS OF MOTION
       F1 = X1*DU + X2*DU*DU + X3*DU*DU*DU + X4*V*V
      1    + X5*R*R + X6*D*D + X7*V*R + X8*V*D + FX
       F2 = YO + Y1*V + Y2*V*V*V + Y3*V*D*D + Y4*R
      1    + Y5*R*V*V + Y6*D + Y7*D*D*D + Y8*D*V*V + FY
       F3 = NO + N1*V + N2*V*V*V + N3*V*D*D + N4*R
      1    + N5*R*V*V + N6*D + N7*D*D*D + N8*D*V*V + MZ
C
       UDOT = F1/C1
       VDOT = (C4*F2-C3*F3)/(C2*C4-C5*C3)
       RDOT = (C2*F3-C5*F2)/(C2*C4-C5*C3)
C   WHEN TO PRINTOUT
       IF (TIME.EQ.0.0) GO TO 50
       IF (ICOUNT.EQ.4) GO TO 50
       GO TO 300
C   CONVERT RADIANS TO DEGREES
   50  YAWDEG= YAW*57.296
       RDEG=R*57.296
       RDDEG=RDOT*57.296
       DDEG=D*57.296
       YAWC=YAWC*57.296
       WRITE (6,100) TIME,U,V,R
  100  FORMAT(' ',1X,F9.2,1X,F9.6,1X,F9.6,1X,F9.6)
       ICOUNT=1
C   TEST IF WANT TO STOP
   300 IF (TIME.GT.ETIME) GO TO 400
C   INTEGRATION STEP SIZE DELT
       DELT=1.
C   INTEGRATION
       U=U+UDOT*DELT
       V=V+VDOT*DELT
       R=R+RDOT*DELT
       YAW=YAW+R*DELT
       S1=S1+DS1*DELT
       S2=S2+DS2*DELT
C**********
       TIME=TIME+DELT
       ICOUNT=ICOUNT+1
       ISE=ISE + LAMDA*YAWE**2
       ISR=ISR + D**2
```

```
      TDIFF=ISE+ISR
      GO TO 200
  400 CONTINUE
      STOP
      END
$ENTRY
$END
```

APPENDIX B

DATA FOR SEA STATE PROGRAM

The sea state program which is explained in [Ref. 4], needs information about the ship to calculate the ship's added mass, added inertia and seaway disturbance forces and moments. Information about the Mariner was gathered from [Ref. 27, 28] and presented here in the form which the sea state program needs. The current line is drawn on next page to show the location of the values in the format

```
...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

   13    13    10     1  32.174  1.98918    -23.9       0.0        8.1
0.1          0.2          0.3        0.4        0.5        0.6       0.75       1.0
1.5          2.5
    7     3     0     1  2.0       15.0        1.         15.0       0.0
0.0          30.0         60.0       90.0       120.0      150.0     180.0
10.0         15.0         20.0
    1     2     3     4     5     6     7     8     9    10
   11    12    13
563.6        76.0         32.0       520.0      22.62      126.68    126.68     0.0
-260.00    -234.00      -208.00    -156.00    -104.00     -52.00    00.00      52.0
104.00     156.00       208.00     234.00     260.00
21093.0                 30090.0    -22.6       2.10       572.0
    1    0.0
   0.0          24.0
   8.53         32.0
    5    0.0
   0.0           0.0
   2.739         4.0
   3.34          8.0
   4.375        16.0
   8.7          24.0
  17.84         32.0
    5    0.0
   0.0           0.0
   6.07          4.0
   7.79          8.0
  11.2          16.0
  18.08         24.0
  25.71         32.0
    6    0.0
   0.0           0.0
   4.43          0.0
  15.03          4.0
  19.28          8.0
  26.187        16.0
  31.77         24.0
  35.61         32.0
    5    0.0
  13.54          0.0
  27.125         4.0
  31.23          8.0
  35.677        16.0
  37.437        24.0
  37.979        32.0
```

```
  5    0.0
24.875      0.0
34.677      4.0
37.073      8.0
37.989     16.0
38.0       24.0
38.0       32.0
  5    0.0
28.479      0.0
36.23       4.0
37.864      8.0
38.0       16.0
38.0       12.50
38.0       16.667
  5    0.0
21.333      0.0
33.25       4.0
35.677      8.0
37.187     16.0
37.635     24.0
37.875     32.0
  5    0.0
 9.166      0.0
24.083      4.0
27.593      8.0
30.864     16.0
32.677     24.0
34.052     32.0
  5    0.0
 2.333      0.0
13.645      4.0
16.78       8.0
19.075     16.0
21.854     24.0
24.156     32.0
  5    0.0
 0.0        0.0
 6.854      4.0
 7.979      8.0
 8.51      16.0
 9.25      24.0
11.083     32.0
  5    0.0
 0.0        0.0
 4.74       4.0
 5.073      8.0
 4.302     16.0
 4.062     24.0
 5.062     32.0
```

```
  5    0.0
0.0          0.0
3.052        4.0
2.687        8.0
1.01        16.0
0.218       24.0
0.416       32.0
```

APPENDIX C

PROGRAM TO CALCULATE OPTIMAL GAINS

Algorithm used here is showed in Figure (C.1).

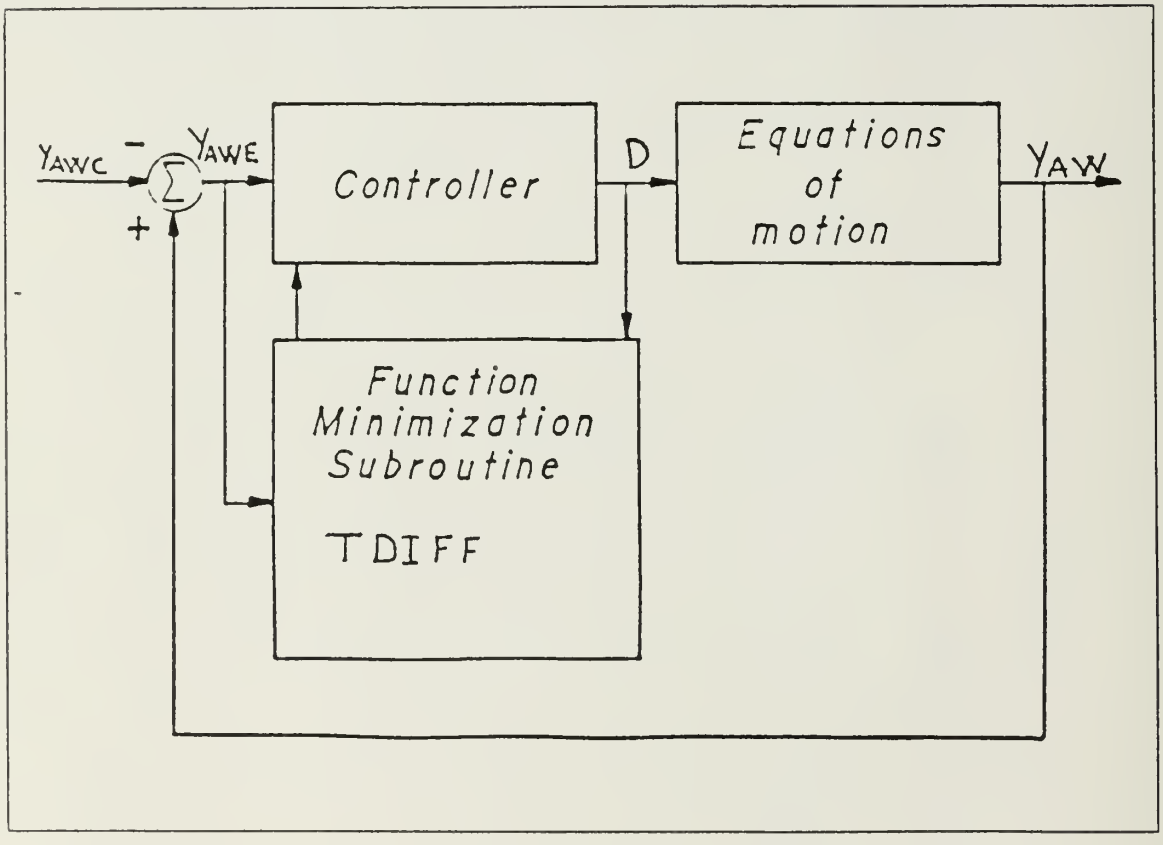


Figure C.1 Algorithm to Calculate Optimal Gains.

The disturbance forces (FX,FY) and moment (MZ) for regular seas were aplied in program as a cosine wave. The procudure to do this is:

- The real and imaginary value of force or moment is read from the sea state program output depending on ship speed, encounter frequency and encounter angle.
- These values are converted to magnitude and phase values.
- Using the formula below, forces and moment are created and added into the ship's equations of motion.

Force or Moment = WA*MAGNITUDE*cos(WE*TIME + PHASE)

Where:

WE = Encounter frequency (radian/second)

WA = Significant wave height (ft)

The correspondence between sea state and significant wave height is indicated in Table XXI [Ref. 29]. During this research the values that are presented in Table XXII were used as significant wave height (WA).

TABLE XXI

Sea State vs Range for Wave Height

Sea State (Beaufort scale)	Range for wave height (Feet)
0	0.0
1	0.32
2	0.65-0.98
3	1.96-3.28
4	3.28-4.92
5	6.56-8.20
6	9.84-13.1
7	13.1-18.2
8	18.2-24.6
9	23.0-32.9

TABLE XXII

Sea State vs Wave Height

Sea state	Wave height (ft)
5	5.0
6	10.0
7	17.5
8	20.0
9	25.0

The program is set up to calculate the optimal gains for controller A. It can be modified to obtain optimal gains for the rest of the controllers. After obtaining the optimal gains the program must be modified to do a simulation. The program has sufficient comments for appropriate changes.

This program can be modified to obtain the Nomoto models. It is referenced in Chapter 4. The following need to be changed.

```
C  GAIN COEFFICIENTS TO BE OPTIMIZED
       K=XX(1)
       Z1=XX(2)
       P1=XX(3)
       P2=XX(4)
C  ERROR SIGNAL TO DRIVE RUDDER (YAW ACTUAL - YAW COMMAND)
c  FOR EQUATIONS OF MOTION.
       D=YAW - YAWC
C  ERROR SIGNAL TO DRIVE RUDDER (YAW COMMAND - YAW ACTUAL)
C  FOR NOMOTO 3RD ORDER MODEL.
       D2=YAWC-YAW2
       DQ1=(D2 - Q1)/P1
       DQ2=((K*((Z*DQ1)+Q1)/P2
C  INTEGRATION
       Q1=Q1+DQ1*DELT
       Q2=Q2+DQ2*DELT
```

```
      YAW2=YAW2+Q2*DELT
C  COST FUNCTION
      TDIFF=TDIFF+ (YAW-YAW2)**2
```

PROGRAM TO CALCULATE OPTIMAL GAINS FOR CONTROLLER

```
//TANSAN JOB (1789,0356),'RESEARCH',CLASS=J
//*MAIN ORG=NPGVM1.1789P
// EXEC FRTXCLGP,IMSL=DP,REGION=1024K
//FORT.SYSIN DD *
C  IN ORDER TO PERFORM SIMULATION ONLY WHEN GAINS HAVE BEEN
C  OBTAINED  CHANGE XS(*)   TO X(*)   AND DELETE  XU(*),AND
XL(*).
      DIMENSION XS(3),XU(3),XL(3)
      XS(1)=0.6
      XS(2)=41.58
      XS(3)=11.33
C  XS(I) IS THE STARTING GUESS
C  XL(I) IS THE LOWER LIMIT FOR THE I'TH VARIABLE
C  XU(I) IS THE UPPER LIMIT FOR THE I'TH VARIABLE
      XL(1)=.3
      XU(1)=0.9
      XL(2)=30.
      XU(2)=50.
      XL(3)=6.
      XU(3)=15.
C  A DESCRIPTION OF THE FOLLOWING PARAMETERS
C  IS DISCUSSED IN BOXPLX
      R=9./13.
      NTA=1000
      NPR=0
      NAV=0
      NV=3
      IP=0
C  THE FOLLOWING STATEMENT MUST BE CHANGED TO
C  CALL PLANT(XX)
C  IF ONLY SIMULATION IS WANTED
      CALL BOXPLX(NV,NAV,NPR,NTA,R,XS,IP,XU,XL,YMN,IER)
      WRITE (6,25)
25    FORMAT(1X,' OPTIMAL GAINS',/)
      DO 30 I=1,3
30    WRITE(6,40)I,XS(I)
40    FORMAT(1X,'X(',I2,')=',F14.7)
      STOP
      END
      SUBROUTINE PLANT(XX)
C  SUBROUTINE PLANT(XX) SIMULATES THE SHIP
      COMMON TDIFF
      REAL*8 L,L2,L3,L4,L5,L6
      REAL*8 X,XDOT,Y,YDOT,U,UDOT,V,VDOT,YAW,R,RDOT
      REAL*8 TIME,ETIME,X1,X2,X3,X4,X5,X6,X7,X8
      REAL*8 Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8
      REAL*8 N0,N1,N2,N3,N4,N5,N6,N7,N8
      REAL*8 C1,C2,C3,C4,C5,F1,F2,F3
      REAL*8 RO,DELT,S,DU,U1,K,Z1,Z2,P1,P2
      REAL*8 DYAWE,YAWE,YAWC,ISR,ISE,TDIFF,LAMDA
      REAL*8 S1,S2,DS1,DS2,D
      REAL*8 MASS,IZ,XG,YVDOT,NVDOT,YR,YRDOT
      REAL*8 NR,NRDOT,FX,FY,MZ,RXR,RYR,RXI,RYI,MZR,MZI
      REAL*8 RX,RY,RZ,TX,TY,TZ,WA,WE
      DIMENSION XX(3)
C  INITIAL CONDITIONS FOR INTEGRATION
C  SIMULATION END TIME IN SECONDS
```

```
      ETIME=600.
      TIME=0.0
      ICOUNT=1
C  INITIALIZE THE COST FUNCTION
      ISE=0.0
      ISR=0.0
      TDIFF=0.0
      LAMDA=2.91
C  GAIN COEFFICIENTS TO BE OPTIMIZED
      K=XX(1)
      Z1=XX(2)
      P1=XX(3)
C  X,XDOT,Y,YDOT ARE FIX COORDINATES ON EARTH
      X=0.0
      Y=0.0
      XDOT=0.0
      YDOT=0.0
C  U,UDOT,V,VDOT ARE FIX COORDINATES ON SHIP
      F1=0.0
      F2=0.0
      F3=0.0
      V=0.0
      UDOT=0.0
      VDOT=0.0
      YAW=0.0
      YAWE=0.0
      R=0.0
      RDOT=0.0
C  ORDERED SPEED IN  FEET/SEC
C   15.*1.689 FT/SEC=15 KNOTS
      U1=15.*1.689
C  AT STEADY STATE ACTUAL SPEED (U) = COMMAND SPEED (UC)
      U=U1
C  D =  RUDDER ANGLE
      D=0.0/57.296
      L=520.
      L2=L**2
      L3=L*L*L
      L4=L*L3
      L5=L*L4
      L6=L*L5
C     FORCES IN X,Y DIRECTION COMPUTED IN FORCES
C     MOMENTS IN Z
      FX=0.
      FY=0.
      MZ=0.
      RXR=.33374D3
      RXI=.16064D3
      RYR=-.17254D5
      RYI=.1435D4
      MZR=.30976D7
      MZI=.84672D6
      RX=(RXR**2+RXI**2)**.5
      RY=(RYR**2+RYI**2)**.5
      RZ=(MZR**2+MZI**2)**.5
      TX=DATAN2(RXI,RXR)
      TY=DATAN2(RYI,RYI)
      TZ=DATAN2(MZI,MZR)
C     SIGNIFICANT WAVE HEIGHT:(SEA STATE 2)
      WA=25.
C     ENCOUNTER FREQUENCY:(WHEN ENCOUNTER ANGLE IS 00)
      WE=2.5
C     ADDED MASS AND ADDED INERTIA TERMS:
      MASS=.14685D+07
      IZ=.23567D+11
      XG=-23.9
      YR=-.24965D8
      YRDOT=.75199D7
      NR=-.47281D10
```

```
      NRDOT=-.29233D11
      YVDOT=-.19723D6
      NVDOT=-.23146D8
C
C HYDRODYNAMIC COEFFICIENTS ARE INSERTED HERE AS PARAMETERS
      RO=1.9876*.5
      YAWE=0.0
      DS1=0.0
      DS2=0.0
      S1=0.0
      S2=0.0
  200 CONTINUE
C  INPUT YAW COMMAND
      YAWC=0.0
C     IF (TIME.GE.0.0) YAWC=(1.0/57.296)
C  ERROR SIGNAL TO DRIVE RUDDER(YAW ACTUAL - YAW ORDERED)
      YAWE=YAW - YAWC
      S=((U*U)+(V*V))**.5
      DU=U-U1
      DS1=(YAWE - S1)/P1
      D=(S1 + DS1*Z1)*K
C  AXIAL FORCE HYDRODYNAMIC COEFFICIENTS (SURGE)
      X1=(-0.0253)*(RO*L2*S)
      X2=(0.00948)*(RO*L2)
      X3=(-0.00217)*(RO*L2/S)
      X4=(-0.189)*(RO*L2)
      X5=(0.00379)*(RO*L4)
      X6=(-0.02)*(RO*L2*S*S)
      X7=(0.168)*(RO*L3)
      X8=(0.0196)*(RO*L2*S)
C  LATERAL FORCE HYDRODYNAMIC COEFFICIENTS (SWAY)
C     Y0=(-0.0008)*(RO*L2*S*S)
      Y0=0.0
      Y1=(-0.244)*(RO*L2*S)
      Y2=(-1.702)*(RO*L2/S)
      Y3=(-0.0008)*(RO*L4*S*S)
C     Y4=(-0.105)*(RO*L3*S)
      Y4=(YR-MASS)
      Y5=(3.23)*(RO*L3/S)
      Y6=(0.0586)*(RO*L2*S*S)
      Y7=(-0.00975)*(RO*L2*S*S)
      Y8=(0.25)*(RO*L2)
C  MOMENT ABOUT Z-AXIS HYDRODYNAMIC COEFFICIENTS (YAW)
C     N0=(0.00059)*(RO*L3*S*S)
      N0=0.0
      N1=(-0.0555)*(RO*L3*S)
      N2=(0.345)*(RO*L3/S)
      N3=(0.00264)*(RO*L3*S)
C     N4=(-0.0349)*(RO*L4*S)
      N4=(NR-MASS*XG*S)
      N5=(-1.158)*(RO*L4/S)
      N6=(-0.0293)*(RO*L3*S*S)
      N7=(0.00482)*(RO*L3*S*S)
      N8=(0.1032)*(RO*L3)
C  COMMON COEFFICIENTS
      C1=(0.177)*(RO*L3)
C     C2=(0.327)*(RO*L3)
      C2=(MASS-YVDOT)
C     C3=(-0.0018)*(RO*L4)
      C3=(MASS*XG-YRDOT)
c     C4=(0.0175)*(RO*L5)
      C4=(IZ-NRDOT)
C     C5=(-0.00478)*(RO*L4)
      C5=(MASS*XG-NVDOT)
C     REGULAR WAVES
      FX=WA*RX*DCOS(WE*TIME+TX)
      FY=WA*RY*DCOS(WE*TIME+TY)
      MZ=WA*RZ*DCOS(WE*TIME+TZ)
C
```

```
      IF(TIME.EQ.0.0) FX=0.0
      IF(DABS(FY).LT.0.00000001) FY=0.0
      IF(DABS(MZ).LT.0.00000001) MZ=0.0
C  EQUATIONS OF MOTION
      F1 = X1*DU + X2*DU*DU + X3*DU*DU*DU + X4*V*V
         + X5*R*R + X6*D*D + X7*V*R + X8*V*D + FX
      F2 = Y0 + Y1*V + Y2*V*V*V + Y3*V*D*D + Y4*R
         + Y5*R*V*V + Y6*D + Y7*D*D*D + Y8*D*V*V + FY
      F3 = N0 + N1*V + N2*V*V*V + N3*V*D*D + N4*R
         + N5*R*V*V + N6*D + N7*D*D*D + N8*D*V*V + MZ
C
      UDOT = F1/C1
      VDOT = (C4*F2-C3*F3)/(C2*C4-C5*C3)
      RDOT = (C2*F3-C5*F2)/(C2*C4-C5*C3)
C  WHEN TO PRINTOUT
      IF (ICOUNT.EQ.21) GO TO 50
      GO TO 300
C  CONVERT RADIANS TO DEGREES
  50  YAWDEG= YAW*57.296
      RDEG=R*57.296
      RDDEG=RDOT*57.296
      DDEG=D*57.296
      YAWC=YAWC*57.296
C     WRITE (6,100) TIME,TDIFF
C 100 FORMAT(' ',1X,F10.2,1X,F20.10)
      ICOUNT=1
C  TEST IF WANT TO STOP
  300 IF (TIME.GT.ETIME) GO TO 400
C  INTEGRATION STEP SIZE DELT
      DELT=1.
C  INTEGRATION
      U=U+UDOT*DELT
      V=V+VDOT*DELT
      R=R+RDOT*DELT
      YAW=YAW+R*DELT
      S1=S1+DS1*DELT
      S2=S2+DS2*DELT
C*********************
      TIME=TIME+DELT
      ICOUNT=ICOUNT+1
      ISE=ISE + LAMDA*YAWE**2
      ISR=ISR + D**2
      GO TO 200
  400 TDIFF=ISE+ISR
      WRITE(6,500) TIME,TDIFF
  500 FORMAT(' ',1X,F9.2,1X,F20.10)
      RETURN
      END
C  DELETE ALL THE FOLLOWING SUBROUTINE IF SIMULATION ONLY
C  AND NOT OPTIMIZATION IS WANTED
C     ..................................................
C
C     SUBROUTINE BOXPLX              (CATEGORY H0)
C
C     PURPOSE
C
C BOXPLX IS A SUBROUTINE USED TO SOLVE THE PROBLEM OF
C LOCATING A MINIMUM (OR MAXIMUM) OF AN ARBITRARY OBJECT-
C IVE FUNCTION SUBJECT TO ARBITRARY EXPLICIT AND/OR
C IMPLICIT CONSTRAINTS BY THE COMPLEX METHOD OF M.J. BOX.
C EXPLICIT CONSTRAINTS ARE DEFINED AS UPPER AND LOWER
C BOUNDS ON THE INDEPENDENT VARIABLES IMPLICIT CONSTRAINTS
C MAY BE ARBITRARY FUNCTION OF THE VARIABLES.  TWO FUN-
C CTION SUBPROGRAM TO EVALUATE THE OBJECTIVE FUNCTION AND
C IMPLICIT CONSTRAINTS, RESPECTIVELY, MUST BE SUPPLIED
C BY THE USER (SEE EXAMPLE BELOW). BOXPLX ALSO HAS tHE
C OPTION TO PERFORM INTEGER PROGRAMMING, WHERE THE VALUES
c OF THE INDEPENDENT VARIABLES ARE RESTRICTED TO INTEGERS.
C
```

```
C     USAGE
C
C        CALL BOXPLX (NV,NAV,NPR,NTA,R,XS,IP,XU,XL,YMN,IER)
C
C
C     DESCRIPTION OF PARAMETERS
C
C NV   AN INTEGER INPUT DEFINING THE NUMBER OF INDEPENDENT
C      VARIABLES OF THE OBJECTIVE FUNCTION TO BE MINIMIZED.
C   NOTE:  MAXIMUM NV + NAV IS PRESENTLY 50.  MAXIMIM NV IS
C     25.  IF THESE LIMITS MUST BE EXCEEDED, PUNCH A SOURCE
C        DECK IN THE USUAL MANNER, AND CHANGE THE DIMENSION
C        STATEMENTS.
C
C NAV AN INTEGER INPUT DEFINING THE NUMBER OF AUXILIARY var
C iaBLES THE USER WISHES TO DEFINE FOR HIS OWN CONVENIENCE.
C TYPICALLY HE MAY WISH TO DEFINE THE VALUE OF EACH IMPLICI
C CONSTRAINT FUNCTION AS AN AUXILIARY VARIABLE.  IF THIS
C IS DONE, THE OPTIONAL OUTPUT FEATURE OF BOXPLX CAN BE
C USED TO OBSERVE THE VALUES OF THOSE CONSTRAINTS AS THE
C SOLUTION PROGRESSES.  AUXILIARY VARIABLES, IF USED,
C SHOULD BE EVALUATED IN FUNCTION KE (DEFINED BELOW).
C NAV MAY BE ZERO.
C
C NPR  INPUT INTEGER CONTROLLING THE FREQUENCY OF OUTPUT
c desired for diagnostic purposes.
C IF NPR .LE. 0, NO OUTPUT WILL BE
C PRODUCED BY BOXPLX.  OTHERWISE, THE CURRENT COMPLEX OF
C K= 2*NV VERTICES AND THEIR CENTROID WILL BE OUTPUT AFTER
C EACH NPR PERMISSIBLE TRIALS.  THE NUMBER OF TOTAL TRIALS,
C NUMBER OF FEASIBLE TRIALS, NUMBER OF FUNCTION EVALUATIONS
C AND NUMBER OF IMPLICIT CONSTRAINT EVALUATIONS ARE IN-
C CLUDED IN THE OUTPUT.
C ADDITIONALLY, (WHEN NPR .GT. 0) THE SAME INFORMATION
C WILL BE OUTPUT:
C
C 1) IF THE INITIAL POINT IS NOT FEASIBLE,
C 2) AFTER THE FIRST COMPLETE COMPLEX IS GENERATED,
C 3) IF A FEASIBLE VERTEX CANNOT BE FOUND AT SOME TRIAL,
C 4) IF THE OBJECTIVE VALUE OF A VERTEX CANNOT BE MADE
C    NO-LONGER-WORST.
C 5) IF THE LIMIT ON TRIALS (NTA) IS REACHED AND,
C 6) WHEN THE OBJECTIVE FUNCTION HAS BEEN UNCHANGED FOR
c 2*NV TRIALS, INDICATING A LOCAL MINIMUM HAS BEEN
C FOUND.
C
C IF THE USER WISHES TO TRACE THE PROGRESS OF A SOLUTION,
C A CHOICE OF NPR = 25, 50 OR 100 IS RECOMMENDED.
C
C NTA  INTEGER INPUT OF LIMIT ON THE NUMBER OF TRIALS
c allowed in the calculation.
C IF THE USER INPUTS NTA .LE. 0, A default
C VALUE OF 2000 IS USED.  WHEN THIS LIMIT IS REACHED
c CONTROL RETURNS TO THE CALLING PROGRAM WITH THE BEST
C ATTAINED OBJECTIVE FUNCTION VALUE IN YMN, AND THE BEST
C ATTAINED SOLUTION POINT IN XS.
C
C R   A REAL NUMBER INPUT TO DEFINE THE FIRST RANDOM NUMBER
C USED IN DEVELOPING THE INITIAL COMPLEX OF 2*NV VERTICIES.
C (0. .GT. R .LT. 1.) IF R IS NOT WITHIN THESE BOUNDS,
C IT WILL BE REPLACED BY 1./3. .
C
C XS   INPUT REAL ARRAY DIMENSIONED AT LEAST NV+NAV.
c the first nv must contain a
C FEASIBLE ORIGIN FOR STARTING THE CAL-
C CULATION.  THE LAST NAV NEED NOT BE INITIALIZED.  UPON
C RETURN FROM BOXPLX, THE FIRST NV ELEMENTS OF THE ARRAY
C CONTAIN THE COORDINATES OF THE MINIMUM OBJECTIVE
C function, AND THE REMAINING NAV (NAV .GE. 0) CONTAIN THe
```

```
C values of THE CORRESPONDING AUXILIARY VARIABLES.
C
C IP   INTEGER INPUT FOR OPTIONAL INTEGER PROGRAMMING.
C if ip=1, THE VALUES OF THE INDEPENDENT VARIABLES WILL
C be replaced WITH INTEGER VALUES (STILL STORED AS REAL*4).
C
C XU   A REAL ARRAY DIMENSIONED AT LEAST NV INPUTTING THE
C upper BOUND ON EACH INDEPENDENT VARIABLE, (EACH EXPLICIT
C conSTRAINT). INPUT VALUES ARE SLIGHTLY ALTERED BY BOXPLX.
C
C XL   A REAL ARRAY DIMENSIONED AT LEAST NV INPUTTING THE
c lower bound on each independent
C VARIABLE, (EACH EXPLICIT CONstraint).
C NOTE:  FOR BOTH XU AND XL CHOOSE REASONABLE
C VALUES IF NONE ARE GIVEN, NOT VALUES WHICH ARE
C magnitudes ABOVE OR BELOW THE EXPECTED SOLUTION.
C input values are SLIGHTLY ALTERED BY BOXPLX.
C
C YMN  THIS OUTPUT IS THE VALUE (REAL*4) OF THE OBJECTIVE
C funcTION,CORRESPONDING TO THE SOLUTION POINT OUTPUT IN XS
C
C IER  INTEGER ERROR RETURN.  TO BE INTERROGATED UPON
C return FROM BOXPLX.  IER WILL BE ONE OF THE FOLLOWING:
C
C =-1  CANNOT FIND FEASIBLE VERTEX OR FEASIBLE CENTROID
C AT THE START OR A RESTART (SEE 'METHOD' BELOW).
C =0   FUNCTION VALUE UNCHANGED FOR 'N' TRIALS.  (WHERE
C N=6*NV+10)  THIS IS THE NORMAL RETURN PARAMETER.
C =1   CANNOT DEVELOP FEASIBLE VERTEX.
C =2   CANNOT DEVELOP A NO-LONGER-WORST VERTEX.
C =3   LIMIT ON TRIALS REACHED.  (NTA EXCEEDED)
C NOTE:  VALID RESULTS MAY BE RETURNED IN ANY OF THE
C        ABOVE CASES.
C
C  EXAMPLE OF USAGE
C
C THIS EXAMPLE MINIMIZES THE OBJECTIVE FUNCTION SHOWN IN
C the EXTERNAL FUNCTION FE(X).  THERE ARE TWO INDEPENDENT
C varIABLES X(1) & X(2), AND TWO IMPLICIT CONSTRAINT
C function X(3) & X(4) WHICH ARE EVALUATED AS AUXILIARY
C variables (see EXTERNAL FUNCTION KE(X) ).
C
C DIMENSION  XS(4),XU(2),XL(2)
CC
CC  STARTING GUESS
C     XS(1) = 1.0
C     XS(2) = 0.5
CC  UPPER LIMITS
C     XU(1) = 6.0
c     XU(2) = 6.0
CC  LOWER LIMITS
C     XL(1) = 0.0
C     XL(2) = 0.0
CC
C     R = 9./13.
C     NTA = 5000
C     NPR = 50
C     NAV = 2
C     NV = 2
C     IP = 0
CC
C     CALL BOXPLX (NV,NAV,NPR,NTA,R,XS,IP,XU,XL,YMN,IER)
C     WRITE(6,1) ((XS(I),I=1,4),YMN,IER)
C    1FORMAT ('1','    THE POINT IS LOCATED AT (XS(I)=) '
c    2,4(e13.7,5x),//,
C    3'    AND THE FUNCTION VALUE IS ',E13.7,' IER = ',I5)
CC
C     STOP
C     END
```

```
CC
CC
C      FUNCTION  KE(X)
C EVALUATE CONSTRAINTS.  SET KE=0 IF NO IMPLICIT CONSTRAINT
C is viOLATED, OR SET KE=1 IF ANY IMPLICIT
c constraint is violated.
C      DIMENSION X(4)
C      X1 = X(1)
C      X2 = X(2)
C      KE = 0
C      X(3) = X1 + 1.732051*X2
C      IF (X(3) .LT. 0. .OR. X(3) .GT. 6.) GO TO 1
C      X(4) = X1/1.732051 -X2
C      IF (X(4) .GE. 0.)  RETURN
CC
C    1 KE = 1
C      RETURN
C      END
CC
CC
C      FUNCTION  FE(X)
C      DIMENSION X(4)
CC
CC  THIS IS THE OBJECTIVE FUNCTION.
C      FE= -(X(2)**3 *(9.-(X(1)-3.)**2)/(46.76538))
C      RETURN
C      END
C
C      METHOD
C
C THE COMPLEX METHOD IS AN EXTENSION AND ADAPTION OF
c the simple method of linear programming.
C STARTING WITH ANY ONE feasible point in n-dimension
C A "COMPLEX" OF 2*N vertices is constructed by
C SELECTING RANDOM POINTS WITHIN THE feasible
C REGION.  FOR THIS PURPOSE N COORDINATES ARE FIRST
C RANDOMLY CHOSEN WITHIN THE SPACE BOUNDED BY EXPLICIT CON-
C STRAINTS.  THIS DEFINES A TRIAL INITIAL VERTEX.
c it is then checked for possible violation
C OF IMPLICIT CONSTRAINTS.  IF one or more are violated,
C THE TRIAL INITIAL VERTEX IS DISPLACED half of its
C DISTANCE FROM THE CENTROID OF PREVIOUSLY SELECTED initial
C VERTICES.  IF NECESSARY THIS DISPLACEMENT PROCESS IS
C REPEATED UNTIL THE VERTEX HAS BECOME FEASIBLE.  IF THIS
c fails to happen after 5*n+10 displacements,
C THE SOLUTION IS ABANDoned. after each vertex is added
C TO THE COMPLEX, THE CURRENT centroid is checked for
C FEASIBILITY.  IF IT IS INFEASIBLE, the last trail
C VERTEX IS ABANDONED AND AN EFFORT TO GENERATE an alter-
C ATIVE TRIAL VERTEX IS MADE.  IF 5*N+10 VERTICES ARE
C ABANDONED CONSECUTIVELY, THE SOLUTION IS TERMINATED.
C
C IF AN INITIAL COMPLEX IS ESTABLISHED, THE BASIC
c computation loop is initiated.
C THESE INSTRUCTIONS FIND THE CURRENT WORST vertex, that
C IS, THE VERTEX WITH THE LARGEST CORRESPONDING value for
C THE OBJECTIVE FUNCTION, AND REPLACE THAT VERTEX BY
C ITS OVER-REFLECTION THROUGH THE CENTROID OF ALL OTHER
c vertices. (if the vertex to be
C REPLACED IS CONSIDERED AS A VECTOR IN n-space,
C ITS OVER-REFLECTION IS OPPOSITE IN DIRECTION, IN-
C CREASED IN LENGTH BY THE FACTOR 1.3, AND COLLINEAR WITH
C THE REPLACED VERTEX AND CENTROID OF ALL OTHER VERTICES.)
C
C WHEN AN OVER-REFLECTION IS NOT FEASIBLE OR REMAINS
c worst, it is considered not-permissible
C AND IS DISPLACED HALFWAY TOWARD the centroid.
C AFTER FOUR SUCH ATTEMPTS ARE MADE UNSUCCESSFULLY
C EVERY FIFTH ATTEMPT IS MADE BY REFLECTING THE OFFENDING
```

```
c vertex through the present best
C VERTEX, INSTEAD OF THROUGH THE CENtroid. if 5*n+10
C DISPLACEMENTS AND OVER-REFLECTIONS OCCUR without a
C SUCCESSFUL (PERMISSIBLE) RESULT, THE CURRENT BEST
C VERTEX IS TAKEN AS AN INITIAL FEASIBLE POINT FOR A
c restart run of the complete process.
C RESTARTING IS ALSO UNDERTAKEN when 6*nv+10 consecutive
C TRIALS HAVE BEEN MADE WITH NO SIGNIFicant change in the
C VALUE OF THE OBJECTIVE FUNCTION.  IN ALL cases,
C RESTARTING IS INHIBITED IF THE LAST RESTART DID NOT
C pRODUCE A SIGNIFICANT IMPROVEMENT IN THE MINIMUM
c attained.
C
C IT IS RECOMMENDED THAT THE USER READ THE REFERENCE FOR
C FURTHER USEFUL INFORMATION.  IT SHOULD BE NOTED THAT THE
C ALGORITHM DEFINED THERE HAS BEEN ALTERED TO FIND THE
C CONSTRAINED MINIMUM, RATHER THAN THE MAXIMUM.
C
C
C
C     REMARKS
C
C THE INTEGER PROGRAMMING OPTION WAS ADDED TO THIS PROGRAM
C AS SUGGESTED IN REFERENCE (2).  A MIXED
c integer/continuous variable version of boxplx
C WOULD BE EASY TO CREATE BY DEclaring "ip" to be an array
C OF NV CONTROL VARIABLES WHERE IP (i)=1 would indicate
C THAT THE I-TH VARIABLE IS TO BE CONFINED to integer
C VALUES.  EACH STATEMENT OF THE FORM 'IF (IP .EQ.1)' etc.
C WOULD THEN NEED TO BE ALTERED TO 'IF (IP(I) .EQ. 1)' etc.
C , WHERE THE SUBSCRIPT IS APPROPRIATELY CHOSEN.  NORMALLY,
C XU AND XL VALUES ARE ALTERED TO BE AN EPSILON 'WITHIN'
c actual values
C DECLARED BY THE USER.  THIS ADJUSTMENT IS NOT MADE
C WHEN IP=1.
C
C NOTE:  NO NON-LINEAR PROGRAMMING ALGORITHM CAN GUARANTEE
c that the answer found is the global
C MINIMUM, RATHER THAN JUST A local minimum. however,
C ACCORDING TO REF.2, THE COMPLEX method has an advantage
C IN THAT IT TENDS TO FIND THE GLOBAL minimum more
C FREQUENTLY THAN MANY OTHER NON-LINEAR PROGRAM-
C MING ALGORITHMS.
C
C IT SHOULD BE NOTED THAT THE AUXILIARY VARIABLE FEATURE
c can also be used to deal with
C PROBLEMS CONTAINING EQUALITY CONstraints. any equality
C CONSTRAINT IMPLIES THAT A GIVEN VARiable is not truly
C INDEPENDENT.  THEREFORE, IN GENERAL, ONE variable
C INVOLVED IN AN EQUALITY CONSTRAINT CAN BE RENUMBERED from
C THE SET OF NV INDEPENDENT VARIABLES AND ADDED TO THE SET
C OF NAV AUXILIARY VARIABLES.  THIS USUALLY INVOLVES
C renumbering THE INDEPENDENT VARIABLES OF THE GIVEN
C problem
C SUBROUTINES AND FUNCTIONS REQUIRED
C
C SUBROUTINE 'BOUT' AND FUNCTION 'FBV' ARE INTEGRAL
C parts of THE BOXPLX PACKAGE.
C
C TWO FUNCTIONS MUST BE SUPPLIED BY THE USER.  THE FIRST,
c ke(x), is used to evaluate the implicit
C CONSTRAINTS.  SET KE=0 AT THE beginning of the function
C , THEN EVALUATE THE IMPLICIT CONstraints. in the example
C ABOVE, THE FIRST CONSTRAINT, X(3), must be within the
C RANGE (0. .LE. X(3) .LE. 6.).  THE SECOND constraint x(4)
C , MUST BE .GE. 0. .  IF EITHER CONSTRAINT IS not within
C THESE BOUNDS, CONTROL IS TRANSFERRED TO STATEMENT 1,
C AND KE IS SET TO "1" AND CONTROL IS RETURNED TO BOXPLX.
C
```

```
C THE SECOND FUNCTION THE USER MUST PROVIDE EVALUATES THE
c objective function. it is
C CALLED FE(X) AS SHOWN IN THE EXAMple above, and fe
C MUST BE SET TO THE VALUE OF THE OBJECTIVE function
C CORRESPONDING TO CURRENT VALUES OF THE NV INDEPENDENT
C VARIABLES IN ARRAY 'X'.
C
C     REFERENCES
C
C BOX, M. J., "A NEW METHOD OF CONSTRAINED OPTIMIZATION
C and a COMPARISON WITH OTHER METHODS",
C computer journal, 8 apr. '65, PP. 45-52.
C
C BEVERIDGE G., AND SCHECHTER R., "OPTIMIZATION: THEORY AND
C PRACTICE", MCGRAW-HILL, 1970.
C
C     PROGRAMMER
C
C        R.R. HILLEARY 1/1966.
C        REVISED FOR SYSTEM 360 4/1967
C        CORRECTED  1/1969
C        REVISED/EXTENDED BY L.NOLAN/R.HILLEARY  2/1975
C        CORRECTED  8/1976
C
C
C     ..............................................
C
  SUBROUTINE BOXPLX (NV,NAV,NPR,NTZ,RZ,XS,IP,BU,BL,YMN,IER)
C
      DIMENSION V(50,50), FUN(50), SUM(25), CEN(25), XS(NV)
     1,bu(nv),bl(NV)
C
      KV = 5
      EP = 1.E-6
      NTA = 2000
      IF (NTZ.GT.0) NTA = NTZ
      R = RZ
      IF (R.LE.0..OR.R.GE.1.) R=1./3.
      NVT = NV+NAV
C
C           TOTAL VARS, EXPLICIT PLUS IMPLICIT
      NT = 0
C           CURRENT TRIAL NO.
      NPT = 0
C           CURRENT NO. OF PERMISSIBLE TRIALS
      NTFS = 0
C        CURRENT NO. OF TIMES F HAS BEEN ALMOST UNCHANGED
CC
CC              CHECK FEASIBILITY OF START POINT
C
      DO 4 I=1,NV
      VT = XS(I)
      IF (BL(I).LE.VT) GO TO 1
      II = -I
      VT = BL(I)
      GO TO 2
    1 IF (BU(I).GE.VT) GO TO 3
      II = I
      VT = BU(I)
    2 IF (NPR.GT.0) WRITE (6,49) II
    3 V(I,1) = VT
      CEN(I) = VT
      IF (IP.EQ.1) GO TO 4
      BL(I) = BL(I)+AMAX1(EP,EP*ABS(BL(I)))
      BU(I) = BU(I)-AMAX1(EP,EP*ABS(BU(I)))
    4 SUM(I) = VT
C
C
```

```
      NCE = 1
C               NUMBER OF CONSTRAINT EVALUATIONS
      I = 1
      IF (KE(V(1,1)).EQ.0) GO TO 5
      IF (NPR.LE.0) GO TO 12
      WRITE (6,50)
      GO TO 12
    5 NFE = 1
C
C   NUMBER OF VERTICES (K) = 2 TIMES NO. OF VARIABLES.
      K = 2*NV
C
C   NUMBER OF DISPLACEMENTS ALLOWED.
      NLIM = 5*NV+10
C
C   NUMBER OF CONSECUTIVE TRIALS WITH UNCHANGED FE TO
C   terminate.
      NCT = NLIM+NV
      ALPHA = 1.3
      FK = K
      FKM = FK-1.
      BETA = ALPHA+1.
C
C   INSURE SEED OF RANDOM NUMBER GENERATOR IS ODD.
      IQR = R*1.E7
      IF (MOD(IQR,2).EQ.0) IQR=IQR+101
C
C                       SET UP INITIAL VERTICES
      FUN(1) = FE(V(1,1))
      YMN = FUN(1)
    6 FI = 1.
      FUNOLD = FUN(1)
C
      DO 15 I=2,K
      FI = FI+1.
      LIMT = 0
    7 LIMT = LIMT+1
C
C   END CALCULATION IF FEASIBLE CENTROID CANNOT BE FOUND.
      IF (LIMT.GE.NLIM) GO TO 11
C
      DO 8 J=1,NV
C
C   RANDOM NUMBER GENERATOR  (RANDU)
      IQR = IQR*65539
      IF (IQR.LT.0) IQR = IQR+2147483647+1
      RQX = IQR
      RQX = RQX*.4656613E-9
      V(J,I) = BL(J)+RQX*(BU(J)-BL(J))
      IF (IP.EQ.1) V(J,I)=AINT(V(J,I)+.5)
    8 CONTINUE
C
      DO 10 L=1,NLIM
      NCE = NCE+1
      IF (KE(V(1,I)).EQ.0) GO TO 13
C
      DO 9 J=1,NV
      VT = .5*(V(J,I)+CEN(J))
      IF (IP.EQ.1) VT = AINT(VT+.5)
      V(J,I) = VT
    9 CONTINUE
C
   10 CONTINUE
C
   11 IF (NPR.LE.0) GO TO 12
      WRITE (6,51) I
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,I,FUN,CEN,I)
   12 IER = -1
      GO TO 48
```

```
C
   13 DO 14 J=1,NV
      SUM(J) = SUM(J)+V(J,I)
   14 CEN(J) = SUM(J)/FI
C
C    TRY TO ASSURE FEASIBLE CENTROID FOR STARTING.
      NCE = NCE+1
      IF (KE(CEN).EQ.0) GO TO 60
      SUM(J) = SUM(J) -V(J,I)
      GO TO 7
   60 NFE = NFE+1
      FUN(I) = FE(V(1,I))
   15 CONTINUE
C
C    END OF LOOP SETTING OF INITIAL COMPLEX.
      IF (NPR.LE.0) GO TO 17
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,CEN,0)
C
C    FIND THE WORST VERTEX, THE 'J'TH.
      J = 1
C
      DO 16 I=2,K
      IF (FUN(J).GE.FUN(I)) GO TO 16
      J = I
   16 CONTINUE
C
C    BASIC LOOP.  ELIMINATE EACH WORST VERTEX IN TURN.
C    it must become NO LONGER WORST, NOT MERELY IMPROVED.
C    find next-to-vertex, THE 'JN'TH ONE.
   17 JN = 1
      IF (J.EQ.1) JN = 2
C
      DO 18 I=1,K
      IF (I.EQ.J) GO TO 18
      IF (FUN(JN).GE.FUN(I)) GO TO 18
      JN = I
   18 CONTINUE
C
C    LIMT = NUMBER OF MOVES DURING THIS TRIAL TOWARD THE
C     centroid DUE TO FUNCTION VALUE.
      LIMT = 1
C
C    COMPUTE CENTROID AND OVER REFLECT WORST VERTEX.
C
      DO 19 I=1,NV
      VT = V(I,J)
      SUM(I) = SUM(I)-VT
      CEN(I) = SUM(I)/FKM
      VT = BETA*CEN(I)-ALPHA*VT
      IF (IP.EQ.1) VT = AINT(VT+.5)
C
C    INSURE THE EXPLICIT CONSTRAINTS ARE OBSERVED.
   19 V(I,J) = AMAX1(AMIN1(VT,BU(I)),BL(I))
C
      NT = NT+1
C
C    CHECK FOR IMPLICIT CONSTRAINT VIOLATION.
C
   20 DO 25 N=1,NLIM
      NCE = NCE+1
      IF (KE(V(1,J)).EQ.0) GO TO 26
C
C    EVERY 'KV'TH TIME, OVER-REFLECT THE OFFENDING VERTEX
C    through the BEST VERTEX.
      IF (MOD(N,KV).NE.0) GO TO 22
      CALL FBV (K,FUN,M)
C
      DO 21 I=1,NV
      VT = BETA*V(I,M)-ALPHA*V(I,J)
```

```
      IF (IP.EQ.1) VT = AINT(VT+.5)
   21 V(I,J) = AMAX1(AMIN1(VT,BU(I)),BL(I))
C
      GO TO 24
C
C    CONSTRAINT VIOLATION:  MOVE NEW POINT TOWARD CENTROID.
C
   22 DO 23 I=1,NV
      VT = .5*(CEN(I)+V(I,J))
      IF (IP.EQ.1) VT = AINT(VT+.5)
      V(I,J) = VT
   23 CONTINUE
C
   24 NT = NT+1
   25 CONTINUE
C
      IER = 1
C
C    CANNOT GET FEASIBLE VERTEX BY MOVING TOWARD CENTROID,
C    OR BY OVER-REFLECTING THRU THE BEST VERTEX.
      IF (NPR.LE.0) GO TO 42
      WRITE (6,52) NT,J
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,CEN,J)
      GO TO 42
C
C    FEASIBLE VERTEX FOUND, EVALUATE THE OBJECTIVE FUNCTION.
   26 NFE = NFE+1
      FUNTRY = FE(V(1,J))
C
C    TEST TO SEE IF FUNCTION VALUE HAS NOT CHANGED.
      AFO = ABS(FUNTRY-FUNOLD)
      AMX = AMAX1(ABS(EP*FUNOLD),EP)
C
C    ACTIVATE THE FOLLOWING TWO STATEMENTS FOR DIAGNOSTIC
c    purposes only.
C                              WRITE                          (6,99)
J,AFO,AMX,FUNTRY,FUNOLD,FUN(J),FUN(JN),NTFS,N
C  99 FORMAT (1X,I3,6E15.7,2I5)
      IF (AFO.GT.AMX) GO TO 27
      NTFS = NTFS+1
      IF (NTFS.LT.NCT) GO TO 28
      IER = 0
      IF (NPR.LE.0) GO TO 42
      WRITE (6,53) K
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,CEN,0)
      GO TO 42
   27 NTFS = 0
C
C    IS THE NEW VERTEX NO LONGER WORST?
   28 IF (FUNTRY.LT.FUN(JN)) GO TO 34
C
C    TRIAL VERTEX IS STILL WORST; ADJUST TOWARD CENTROID.
C    EVERY 'KV'TH TIME, OVER-REFLECT THE OFFENDING VERTEX
C    through the BEST VERTEX.
      LIMT = LIMT+1
      IF (MOD(LIMT,KV).NE.0) GO TO 30
      CALL FBV (K,FUN,M)
C
      DO 29 I=1,NV
      VT = BETA*V(I,M)-ALPHA*V(I,J)
      IF (IP.EQ.1) VT = AINT(VT+.5)
   29 V(I,J) = AMAX1(AMIN1(VT,BU(I)),BL(I))
C
      GO TO 32
C
   30 DO 31 I=1,NV
      VT = .5*(CEN(I)+V(I,J))
      IF (IP.EQ.1) VT = AINT(VT+.5)
      V(I,J) = VT
```

```
   31 CONTINUE
C
   32 IF (LIMT.LT.NLIM) GO TO 33
C
C    CANNOT MAKE THE 'J'TH VERTEX NO LONGER WORST BY
c    displacing toward
C THE CENTROID OR BY OVER-REFLECTING THRU THE BEST VERTEX.
      IER = 2
      IF (NPR .LE. 0)  GO TO 42
      WRITE (6,52)  NT, J
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,CEN,J)
      GO TO 42
   33 NT = NT+1
      GO TO 20
C
C    SUCCESS:  WE HAVE A REPLACEMENT FOR VERTEX J.
   34 FUN(J) = FUNTRY
      FUNOLD = FUNTRY
      NPT = NPT+1
C
C    EVERY 100'TH PERMISSIBLE TRIAL, RECOMPUTE CENTROID
C    summation to AVOID CREEPING ERROR.
      IF (MOD(NPT,100).NE.0) GO TO 37
C
      DO 36 I=1,NV
      SUM(I) = 0.
C
      DO 35 N=1,K
   35 SUM(I) = SUM(I)+V(I,N)
C
      CEN(I) = SUM(I)/FK
   36 CONTINUE
C
      LC = 0
      GO TO 39
C
   37 DO 38 I=1,NV
   38 SUM(I) = SUM(I)+V(I,J)
C
      LC = J
C
   39 IF (NPR.LE.0) GO TO 40
      IF (MOD(NPT,NPR).NE.0) GO TO 40
C
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,CEN,LC)
C
C    HAS THE MAX. NUMBER OF TRIALS BEEN REACHED WITHOUT
C    convergence? iF NOT, GO TO NEW TRIAL.
   40 IF (NT.GE.NTA) GO TO 41
C
C    NEXT-TO-WORST VERTEX NOW BECOMES WORST.
      J = JN
      GO TO 17
   41 IER = 3
      IF (NPR.GT.0) WRITE (6,54)
C
C    COLLECTOR POINT FOR ALL ENDINGS.
C   1)  CANNOT DEVELOP FEASIBLE VERTEX.              IER = 1
C   2)  CANNOT DEVELOP A NO-LONGER-WORST VERTEX.     IER = 2
C   3)  FUNCTION VALUE UNCHANGED FOR K TRIALS.       IER = 0
C   4)  LIMIT ON TRIALS REACHED.                     IER = 3
C   5)  CANNOT FIND FEASIBLE VERTEX AT START.        IER = -1
   42 CONTINUE
C
C    FIND BEST VERTEX.
      CALL FBV (K,FUN,M)
      IF (IER.GE.3) GO TO 44
C
C    RESTART IF THIS SOLUTION IS SIGNIFICANTLY BETTER THAN
```

```
C    the previous, OR IF THIS IS THE FIRST TRY.
        IF (NPR.LE.0) GO TO 43
        WRITE (6,55) (M,YMN,FUN(M))
     43 IF (FUN(M).GE.YMN) GO TO 47
        IF (ABS(FUN(M)-YMN).LE.AMAX1(EP,EP*YMN)) GO TO 47
C
C    GIVE IT ANOTHER TRY UNLESS LIMIT ON TRIALS REACHED.
     44 YMN = FUN(M)
        FUN(1) = FUN(M)
C
        DO 45 I=1,NV
        CEN(I) = V(I,M)
        SUM(I) = V(I,M)
     45 V(I,1) = V(I,M)
C
        DO 46 I=1,NVT
     46 XS(I) = V(I,M)
C
        IF (IER.LT.3) GO TO 6
     47 IF (NPR.LE.0) GO TO 48
        CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,V(1,M),-1)
        WRITE (6,56) FUN(M)
     48 RETURN
C
      49 FORMAT (50H0INDEX AND DIRECTION OF OUTLYING
        1variable at starti5)
      50 FORMAT (50H0IMPLICIT CONSTRAINT VIOLATED AT
        1start. dead end.)
     51 FORMAT ('0CANNOT FIND FEASIBLE',I4,'TH VERTEX OR
       1centroid at start.')
     52 FORMAT (10H0AT TRIAL I4,54H CANNOT FIND FEASIBLE
       1vertex which is no
       2LONGER WORST,I4,15X,'RESTART FROM BEST VERTEX.')
     53 FORMAT (40H0FUNCTION HAS BEEN ALMOST UNCHANGED
       1for i5,7h trails)
     54 FORMAT (27H0LIMIT ON TRIALS EXCEEDED. )
     55 FORMAT('0BEST VERTEX IS NO.',I3,'OLD MIN WAS',E15.7,
       1 '   NEW MIN IS ',E15.7)
     56 FORMAT ('0MIN OBJECTIVE FUNCTION IS ',E15.7)
        END
        SUBROUTINE FBV (K,FUN,M)
        DIMENSION FUN(50)
        M = 1
C
        DO 1 I=2,K
        IF (FUN(M).LE.FUN(I)) GO TO 1
        M = I
      1 CONTINUE
C
        RETURN
        END
        SUBROUTINE BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FN,C,IK)
        DIMENSION V(50,50), FN(50), C(25)
        WRITE (6,4) NT,NPT,NFE,NCE
C
        DO 1 I=1,K
        WRITE (6,5) FN(I),(V(J,I),J=1,NV)
        IF (NVT.LE.NV) GO TO 1
        NVP = NV+1
        WRITE (6,6) (V(J,I),J=NVP,NVT)
      1 CONTINUE
C
        IF (IK.NE.0) GO TO 2
C
        WRITE (6,7) (C(I),I=1,NV)
        RETURN
      2 IF (IK.GE.0) GO TO 3
        WRITE (6,8) (C(I),I=1,NV)
        RETURN
```

```
     3 WRITE (6,9) IK,(C(I),I=1,NV)
       RETURN
C
     4 FORMAT ('0NO. TOTAL TRIALS = ',I5,4X,
      1'no. feasible trails = ',i5,4x,
      2'NO. FUNCTION EVALUATIONS = ',I5,4X,
      3'no. constraint evaluations = ',i5/
      4'0     FUNCTION VALUE',6X,'INDEPENDENT VARIABLES/
      5dependENT OR IMPLICIT CONSTRAINTS')
     5 FORMAT (1H ,E18.7,2X,7E14.7/(21X,7E14.7))
     6 FORMAT (21X,7E14.7)
     7 FORMAT (10HOCENTROID 11X,7E14.7/(21X,7E14.7))
     8 FORMAT ('0  BEST VERTEX',7X,7E14.7/(21X,7E14.7))
     9 FORMAT ('0CENTROID LESS VX',I2,2X,7E14.7/(21X,
      17e14.7))
       END
       FUNCTION FE(X)
       DIMENSION X(3)
       COMMON TDIFF
       CALL PLANT(X)
       FE=TDIFF
       RETURN
       END
       FUNCTION KE(X)
       DIMENSION X(3)
       KE=0
       RETURN
       END
//GO.SYSIN DD *
/*
C THIS CARD SHOULD BE USED ONLY REGULAR SEA CASE
//GO.FT12F001 DD DISP=SHR,DSN=MSS.S2160.A341
```

LIST OF REFERENCES

1. Abkowitz M. A., Lectures in Ship Hydrodynamics, Steering and Maneuverability, Hydro-Og Aerodynamisk Laboratorium, Report Hy-5, Lyngby, Copenhagen, Denmark, May 1964.

2. Abkowitz, M.A., Stability and Motion Control of Ocean Vehicles, Massaachustts Institute of Technology, National, Science Foundation, Sea Grant Project GH-1 May 1969.

3. Strom-Tejsen, J., A Digital Computer Technique for the Prediction of Standard Maneuvers of Surface Ships, DTMB Report 2130, December 1965.

4. LTCDR, Cass James, U.S Navy, Theory and Application of a Sea State Simulation Program, M.S. Thesis, Naval Postgraduate School, Monterey,CA, September 1984.

5. Reid, R.E. , Improvement of Ship Steering Control for Merchant Ships, Phase IIa, National Maritime Research Center, Kings Point, New York, October 1978. Reid, R.E. and Moore J.W.,

6. LTCDR, Vincente Garcia, U.S Navy, Optimization of Guidance and Control Using Function Minimization and NAVSTAR/GPS, M.S. Thesis, Naval Postgraduate School, Monterey, CA, September, 1984.

7. Horianopoulas, E. Optimization of Surface Ship Steering in Sea State, M.S. Thesis, Naval Postgraduate School, Monterey, CA, December 1984.

8. Spyromilios, P.K. Development of Criteria for Automatic Steering, M.S. Thesis, Naval Postgraduate School, Monterey, CA, December 1984.

9. Vincenti, G.C. and Thaler, G.J., Comparison of steering Control Algorithms for Optimized Autopilots, Seventh Ship Control Systems Symposium, Bath,United Kingdom, 27 September 1984.

10. Milliken, R.J. and Zoller, C.J., "Principle of Operation of NAVSTAR and System Characteristics," Navigation: Journal of The Institute of Navigation, Vol 25, No. 2, Summer 1978.

11. Van Dierendonck, A.,J., Russell, S.S., Kopitzke, E.R., and Birnbaum, M., "The GPS Navigation Message," Navigation: Journal of The Institute of Navigation, Vol 25, No. 2, Summer 1978.

12. Jorgensen, P.S., "NAVSTAR/Global Positioning System 18-Satellite Constellations," Navigation: Journal of The Institute of Navigation, Vol 27, No. 2, 1980.

13. Esaias, W.E., Greaves, J.R., Patzert, W.C., Thomas, R.H., Townsend, W.F. and Wilson W.S., NASA Oceanic Processes Program, NASA Technical Memorandum 86248, pp II-17 and II-18, May 1984.

14. Cowley W.E. and Lamber T.H., Sea Trials on a Roll Stabiliser Using the Ship's Rudder, Department of Mechanical Engineering, University Collage London. Seventh Ship Control Systems Symposium 1984.

15. A R J M Lloyd PhD BSc MRINA, Roll Stabilisation by Rudder, Admiralty Experiment Works. Seventh Ship Control Systems Symposium 1984.

16. Nomoto, K. and Motoyama, T., "Loss of Propulsive Power Caused by Yawing with Particular Reference to automatic Steering", Japan Shipbuilding and Marine Engineering Vol. 120, 1966.

17. Motora, S. and Koyama, T., "Some Aspects of Automatic Steering of Ships", Japan Shipbuilding and Marine Engineering., Vol. 3 No. 4 july 1968.

18. Norrbin, N.H., On the Added Resistance due to Steering on a Straight Course, 13th ITTc Report of Performance Committee 1972.

19. Bech, M.I., "Some Aspects of the Stability of Automatic Course Control of Ships", J. of Mech. Engineering Science, Vol. 14 No. 7 1972.

20. van Amerongen, J. and van Nauta lemke, H.r., Optimum Steering of Ships with an Adaptive Autopilots, Proc 5th Ship Control System Symposium, Annapolis, Md 1978.

21. Kallstrom, C.G., "et al Adaptive Autopilots for Tankers", Automatica Vol. 15 1979.

22. Astom, K.J., Design of Fixed Gain and Adaptive Autopilots based on the Nomoto Model, Proc of Symposium on Ship Steering Automatic Control, Genoav, Italy, June 1980.

23. Reid, R.E. and Moore J.W., Optimal Steering Control of High Speed Containership, Part I System Dynamics, Part II Controller Design Proc 1980 ACC. San Francisco, Ca, August 1980.

24. van Amerongen, V. and van Nauta Lemke, H.R., Criteria for OPtimum Steering of Ships, Proceedings of Symposium on Ship Steering Automatic Control, Genova, Italy , June 1980.

25. Kallstrom, C.G. and Norrbin, N.H., Performance Criteria for Ship Autopilots: An Analysis of Shipboard Experiments, Proceedings of Symposium on Ship Steering Automatic Control, Genova, Italy 1980.

26. Clarke, D., Development of a Cost Function for Autopilot Optimization, Proceedings of Symposium on Ship Steering Automatic Control, Genova, Italy 1980.

27. Comstock, J.P., Principles of Naval Architecture, The Society of Naval Architects and Marine Engineers, 1967.

28. Dalzell, J.F., An Investigation of Midship Bending Moments Experienced in Extreme Regular Waves by Models of the Mariner Type Ship and Three Variants, Stevens Institude of Technology, JAN, 1964

29. Meteorology for Mariners, Her Majesty's Stationery Office, London, 1978

INITIAL DISTRIBUTION LIST

No.		Copies
1.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
2.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
3.	Professor George J. Thaler, Code 62Tr Department of Electrical and Computer Engineering, Naval Postgraduate School Monterey, California 93943	5
4.	Professor Alex Gerba Jr., Code 62Gz Department of Electrical and Computer Engineering, Naval Postgraduate School Monterey, California 93943	1
5.	Department Chairman, Code 62Rr Department of Electrical and Computer Engineering, Naval Postgraduate School Monterey, California 93943	1
6.	Mr George Curry HENSCHEL 9 Hoyt Drive, P.O. BOX 30 Newburyport, Mass 01950	1
7.	Mr John Carter HENSCHEL 9 Hoyt Drive, P.O. BOX 30 Newburyport, Mass 01950	1
8.	LT. Garcia Vicente, USN P.O. Box 985 Anthony, New Mexico 88021	1
9.	Tayfun Tansan Goztepe Istasyon cad. No: 31/13 Istanbul, Turkey	5
10.	LT. Kyritsis-Spyromilios Pericles, H.N. Sarantaporou 125, Papagou Athens, Greece	1
11.	LT. Horianopoulos Emmanuel, H.N. Travlantoni 39, Patisia Athens, Greece	1
12.	Dz. K. Komutanligi Okullar ve Kurslar Dairesi Bakanliklar, Ankara, Turkey	1

213143

Thesis
T13732 Tansan
c.1 An automatic control design for the Mariner Class Ships.

11 OCT 88 33588

213143

Thesis
T13732 Tansan
c.1 An automatic control design for the Mariner Class Ships.

AMERICAN BINDERIES 2

An automatic control design for the Mari
3 2768 000 61453 1
DUDLEY KNOX LIBRARY